A Machine Learning Perspective on Repeated Measures: Gaussian Process Panel and Person-Specific EEG Modeling

Dissertation
zur Erlangung des akademischen Grades
Doctor rerum naturalium
(Dr. rer. nat.)

eingereicht an der Lebenswissenschaftlichen Fakultät der
Humboldt-Universität zu Berlin

von Dipl.-Inform. Julian David Karch

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. habil. Dr. Sabine Kunst

Dekan der Lebenswissenschaftlichen Fakultät:
Prof. Dr. rer. nat. Dr. Richard Lucius

Gutachter

1. Prof. Dr. Manuel C. Voelkle
2. Prof. Dr. Steven M. Boker
3. Prof. Dr. Thad Polk

Tag der mündlichen Prüfung: 10.10.2016

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt,

- dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte Hilfe verfasst habe,
- dass ich mich nicht bereits anderwärts um einen Doktorgrad beworben habe und keinen Doktorgrad in dem Promotionsfach Psychologie besitze,
- dass ich die zugrunde liegende Promotionsordnung vom 11. Februar 2015 kenne.

Berlin, den 28.06.2016

Julian D. Karch

Acknowledgements

This dissertation is the outcome of research that I conducted within the research project "Formal Methods in Lifespan Psychology" at the Center for Lifespan Psychology of the Max Planck Institute for Human Development in Berlin. During my dissertation work, I also became affiliated with the "Psychological Research Methods" chair at the Humboldt University Berlin. Thus, many people have contributed to the success of this thesis.

Especially, I want to thank:

Andreas Brandmaier for his excellent mentoring throughout this thesis. In particular, his constant availability for discussions has far exceeded what can be expected from a thesis advisor. To this day, I am still surprised how fast problems can be resolved when discussing them with him.

Manuel Völkle for his excellent co-mentoring of the thesis. He always gave important support and help.

Ulman Lindenberger, the director of the Center for Lifespan Psychology, for always being supportive of my work.

Markus Werkle-Bergner, Myriam Sander, and *Timo von Oertzen* for co-authoring the first paper that resulted as part of my dissertation work.

Timo von Oertzen also for convincing me in the first place that the Center for Lifespan Psychology is a great place to work, and for his continuing methodological advice.

Janne Adolf and *Charles Driver*, my fellow PhD students within the Formal Methods project, for countless hours of helpful discussion.

Janne Adolf also for proofreading the introduction, the discussion, and Chapter 2, and for repeatedly suggesting valuable literature.

Steven Boker, *Thad Polk*, *Matthias Ziegler*, and *Martin Hecht*, my committee members, for being willing to invest time and effort in this thesis.

Julia Delius for providing highly valuable editorial assistance.

Florian Schmiedek for providing the COGITO data set.

Contributions

The work presented in Chapter 5 has already been published:

Karch, J. D., Sander, M. C., von Oertzen, T., Brandmaier, A. M., & Werkle-Bergner, M. (2015). Using within-subject pattern classification to understand lifespan age differences in oscillatory mechanisms of working memory selection and maintenance. *NeuroImage*, *118*, 538–552. doi:10.1016/j.neuroimage.2015.04.038

Here, I embed this project within the remaining work of my dissertation, and focus on the methodological aspects. My contributions for the original publication were:

Programming: entirely; data analysis: entirely; compiling of the manuscript: largely; development of methods: largely; reasoning: largely; literature research: largely; discussion of the results: predominantly; original idea: partially.

Zusammenfassung

Wiederholte Messungen mehrerer Individuen sind von entscheidender Bedeutung für die entwicklungspsychologische Forschung. Nur solche Datenstrukturen erlauben die notwendige Trennung von Unterschieden innerhalb von und Unterschieden zwischen Personen. Beispiele sind längsschnittliche Paneldaten und Elektroenzephalografie-Daten (EEG-Daten). In dieser Arbeit entwickle ich für jede dieser beiden Datenarten neue Analyseansätze, denen Methoden des maschinellen Lernens zu Grunde liegen.

Für Paneldaten entwickle ich Gauß-Prozess-Panelmodellierung (GPPM), die auf der flexiblen Bayesschen Methode der Gauß-Prozess-Regression basiert. Damit GPPM dem psychologischen Fachpublikum zugänglich wird, leite ich außerdem begleitende frequentistische Inferenzverfahren her. Der Vergleich von GPPM mit längsschnittlicher Strukturgleichungsmodellierung (SEM), welche die meisten herkömmlichen Panelmodellierungsmethoden als Sonderfälle enthält, zeigt, dass längsschnittliche SEM wiederum als Sonderfall von GPPM aufgefasst werden kann. Im Gegensatz zu längsschnittlicher SEM eignet sich GPPM gut zur zeitkontinuierlichen Modellierung, kann eine größere Menge von Modellen beschreiben, und beinhaltet einen einfachen Ansatz zur Generierung personenspezifischer Vorhersagen. Wie ich ebenfalls zeige, stellt auch die zeitkontinuierliche Modellierungstechnik der Zustandsraummodellierung – trotz vieler Unterschiede – einen Spezialfall von GPPM dar. Ich demonstriere die Vielseitigkeit von GPPM anhand zweier Datensätze und nutze dazu die eigens entwickelte GPPM-Toolbox. Für ausgewählte populäre längsschnittliche Strukturgleichungsmodelle zeige ich, dass die implementierte GPPM-Darstellung gegenüber bestehender SEM Software eine bis zu neunfach beschleunigte Parameterschätzung erlaubt.

Für EEG-Daten entwickle ich einen personenspezifischen Modellierungsansatz zur Identifizierung und Quantifizierung von Unterschieden zwischen Personen, die in konventionellen EEG-Analyseverfahren ignoriert werden. Im Rahmen dieses Ansatzes wird aus einer großen Menge hypothetischer Kandidatenmodelle das beste Modell für jede Person ausgewählt. Zur Modellauswahl wird ein Verfahren aus dem Bereich des maschinellen Lernens genutzt. Als Kandidatenmodelle werden Vorhersagefunktionen verwendet, die die EEG-Daten mit Verhaltensdaten verbinden. Im Gegensatz zu klassischen Anwendungen maschinellen Lernens ist die Interpretation der ausgewählten Modelle hier von entscheidender Bedeutung. Aus diesem Grund zeige ich, wie diese sowohl auf der Personen- als auch auf der Gruppenebene interpretiert werden können. Ich validiere den vorgeschlagenen Ansatz anhand von Daten zur Arbeitsgedächtnisleistung. Die Ergebnisse verdeutlichen, dass die erhaltenen personenspezifischen Modelle eine genauere Beschreibung des Zusammenhangs von Verhalten und Hirnaktivität ermöglichen als konventionelle, nicht personenspezifische EEG-Analyseverfahren.

Abstract

Repeated measures obtained from multiple individuals are of crucial importance for developmental research. Only they allow the required disentangling of differences between and within persons. Examples of repeated measures obtained from multiple individuals include longitudinal panel and electroencephalography (EEG) data. In this thesis, I develop a novel analysis approach based on machine learning methods for each of these two data modalities.

For longitudinal panel data, I develop Gaussian process panel modeling (GPPM), which is based on the flexible Bayesian approach of Gaussian process regression. For GPPM to be accessible to a large audience, I also develop frequentist inference procedures for it. The comparison of GPPM with longitudinal structural equation modeling (SEM), which contains most conventional panel modeling approaches as special cases, reveals that GPPM in turn encompasses longitudinal SEM as a special case. In contrast to longitudinal SEM, GPPM is well suited for continuous-time modeling, can express a larger set of models, and includes a straightforward approach to obtain person-specific predictions. The comparison of GPPM with the continuous-time modeling technique multiple-subject state-space modeling (SSM) reveals, despite many differences, that GPPM also encompasses multiple-subject SSM as a special case. I demonstrate the versatility of GPPM based on two data sets. The comparison between the developed GPPM toolbox and existing SEM software reveals that the GPPM representation of popular longitudinal SEMs decreases the amount of time needed for parameter estimation up to ninefold.

For EEG data, I develop an approach to derive person-specific models for the identification and quantification of between-person differences in EEG responses that are ignored by conventional EEG analysis methods. The approach relies on a framework that selects the best model for each person based on a large set of hypothesized candidate models using a model selection approach from machine learning. Prediction functions linking the EEG data to behavior are employed as candidate models. In contrast to classical machine learning applications, interpretation of the selected models is crucial. To this end, I show how the obtained models can be interpreted on the individual as well as on the group level. I validate the proposed approach on a working memory data set. The results demonstrate that the obtained person-specific models provide a more accurate description of the link between behavior and EEG data than the conventional nonspecific EEG analysis approach.

Contents

Acronyms

AIC Akaike information criterion

ANOVA analysis of variance

AR autoregressive

BAC balanced accuracy

BCI brain–computer interface

BIC Bayesian information criterion

CI confidence interval

CSP common spatial pattern

DV dependent variable

EEG electroencephalography

ERP event-related potential

GLM general linear model

GP Gaussian process

GPML Gaussian processes for machine learning

GPR Gaussian process regression

GPPM Gaussian process panel modeling/model

GPTSM Gaussian process time series modeling/model

HLM hierarchical linear modeling/model

ICA independent component analysis

iff if, and only if,

iid independent and identically distributed

IV independent variable

LGCM latent growth curve model

MAP maximum a posteriori

MLR multiple linear regression

ML maximum likelihood

MRI magnetic resonance imaging

LDA linear discriminant analysis

LLDA Ledoit's linear discriminant analysis

PCA principal component analysis

pdf probability density function

RAM reticular action model

SDE stochastic differential equation

SEM structural equation modeling/model

SSM state-space modeling/model

WM working memory

1. Introduction

Repeated measures obtained from multiple individuals potentially on multiple variables are of crucial importance for developmental research. Only such data allow implementation of the corresponding rationales put forward by Baltes and Nesselroade (1979). These encompass: the "direct identification of intra-individual change [and variation]" (Baltes & Nesselroade, 1979, p. 23), the "direct identification of inter-individual differences" therein (Baltes & Nesselroade, 1979, p. 24), and the "analysis of interrelationships of ... change" (Baltes & Nesselroade, 1979, p. 26). To analyze such data, a model for inter-individual as well as intra-individual variation is required. In this thesis, I adopt a machine learning perspective on modeling repeated measures data. Besides recontextualizing traditional inference methods for model selection and validation, this mainly involves proposing two novel modeling approaches, each tailored to a specific instance of multiple individuals' repeated measures data. For longitudinal panel data, I propose Gaussian process panel modeling (GPPM), and for EEG data, I propose a method to derive person-specific models. Both approaches extend conventional modeling approaches. Specifically, they are not subject to assumptions rarely met for psychological phenomena and thus model multiple individuals' repeated measures data more appropriately.

1.1. Gaussian Process Panel Modeling

For longitudinal panel studies (referred to as panel studies in the following), the most widely used method for formulating a model of intra-individual variation is the *general linear model* (GLM) with chronological time as *independent variable* (IV) and the characteristic of interest as *dependent variable* (DV). Often, other than time, further IVs that are hypothesized to explain both intra-individual and inter-individual variation are included. To allow for inter-individual variation that is not explained by the IVs, the GLM is typically extended to the *hierarchical linear model* (HLM), in which, in contrast to the GLM, the mapping from the IVs to the DV can vary between persons. A prominent example of a HLM for panel data is the *latent growth curve model* (LGCM) in which linear growth rates are assumed to vary between persons.

Structural equation modeling (SEM) is a modeling technique that generalizes all linear, multivariate statistical models such as the t-test, paired t-test, or analysis of variance (ANOVA) (Fan, 1997; Knapp, 1978). It can be shown that the HLM is also a special case of SEM (Curran, 2003). SEM is widely used to model panel data. In this thesis, *longitudinal SEM* will refer to using SEM for panel data.

Model specification in SEM consists of formulating a set of linear equations that describe the hypothesized relationships between a set of variables. Accordingly, it suffers

from two major limitations: First, only linear relationships between variables can be modeled, and second, the number of variables is necessarily finite. The second limitation is especially problematic for panel data. It leads to the problem that longitudinal SEM only allows expression of discrete-time models. Thus, longitudinal SEM does not permit to model development over time as what it is, namely a continuous-time process. To model continuous-time processes, it is necessary to employ continuous-time modeling.

Besides this conceptual advantage, continuous-time modeling also has the advantage that it does not rely on assumptions that can seldom be met in practice. Using discrete-time models, it is assumed that all measurements have been taken at the same time points for all individuals, and that the interval between any two successive measurements is the same. This assumption is rarely met in actual panel studies, in which an occasion of measurement needs to be spread over weeks or months, typically due to limited testing facilities.

One remedy to the problem of longitudinal SEM only being able to express discrete-time models is to use an underlying continuous-time model and generate the corresponding discrete-time structural equation model (SEM) for a given data set. For LGCM this can be straightforwardly achieved. Voelkle, Oud, Davidov, and Schmidt (2012) use this approach to derive a SEM representation of the continuous-time *autoregressive* (AR) model. However, their solution relies on extensions of the conventional SEM approach. The resulting modeling technique is known as extended SEM (Neale et al., 2016), which captures a broader scope of models. This solution is in the spirit of fitting available tools to new problems.

In this thesis, I propose a different approach. Instead of using SEM or extensions of it, I suggest that a more suitable technique that is able to represent continuous-time models directly should be employed. To this end, I systematically introduce the continuous-time time series modeling technique *Gaussian process time series modeling* (GPTSM) as a novel tool for statistical modeling of psychological time series data.

GPTSM is based on *Gaussian process regression* (GPR), which is a flexible function-fitting approach. GPR has recently gained popularity in the field of machine learning as a Bayesian nonparametric regression technique. GPTSM has already been used in fields such as machine learning (Cunningham, Ghahramani, & Rasmussen, 2012; Saatçi, Turner, & Rasmussen, 2010) and physics (Roberts et al., 2013). Within psychology, I am only aware of two applications of GPTSM (Cox, Kachergis, & Shiffrin, 2012; Ziegler, Ridgway, Dahnke, & Gaser, 2014). Both applications adapt GPTSM for specific problems rather than introducing and discussing the method for psychological research in a broader context.

I extend the time series method GPTSM for use on panel data. A panel data set can be interpreted as consisting of a time series for each person. Thus, to extend the time series method GPTSM to panel data, a mechanism allowing formulation of a simultaneous model for multiple time series needed to be composed. I develop one straightforward and unified approach, and call the resulting model family *Gaussian process panel models* (GPPMs). In principle, the Bayesian inference methods used for *Gaussian process time series models* (GPTSMs) could also be used for GPPMs. However, within psychology

frequentist inference is still the de-facto standard approach. Thus, for the method to be interpretable within common statistical reference frames and to be accessible to a large audience, I also develop frequentist inference procedures for GPPMs, and call the resulting method *Gaussian process panel modeling* (GPPM).

While an extension of GPTSM for panel data has already been discussed briefly within the field of statistics (Hall, Müller, & Yao, 2008), a full presentation of a panel modeling approach based on GPTSM, with a full set of corresponding inference procedures and a detailed comparison with conventional panel methods, as I perform here, is still lacking. Also, GPPM is substantially different then the approach proposed by Hall et al. (2008). One main difference is that I propose using parametric estimators, whereas Hall et al. (2008) propose applying nonparametric estimators. To the best of my knowledge, my work is the first work to discuss the extension of GPTSM for modeling of psychological panel data.

Besides GPTSM, other continuous-time modeling techniques could have been used as a basis for a panel modeling framework. Indeed, Oud and Singer (2008), Boker (2007a) have extended the continuous-time time series method *state-space modeling* (SSM) for panel data (i.e., for multiple subjects). I will come back to the difference between multiple-subject SSM and GPPM later.

In this thesis, I examine the properties of GPPM in detail. Specifically, I compare GPPM against longitudinal SEM and multiple-subject SSM. Interestingly, this comparison reveals that both longitudinal SEM and multiple-subject SSM can be regarded as special cases of GPPM. At the same time, GPPM extends both methods. I provide examples of models that can only be expressed using GPPM, and show that they are viable alternatives to related models that are typically used in psychological research. One interesting difference between the continuous-time modeling methods multiple-subject SSM and GPPM is that in GPPM the to-be-modeled process is described explicitly as a function of the IVs, whereas in multiple-subject SSM the process is described implicitly via the dynamics of the process.

In this thesis, I also examine whether expressing longitudinal models as GPPM and using GPPM software to estimate parameters is faster then using the equivalent longitudinal SEM. My findings suggest that GPPM software is indeed faster, and, thus, the use of GPPM software for conventional SEM may either allow the use of larger-scale models or a significant increase of fitting speed.

1.2. Person-Specific EEG Modeling

Since GPPM is a parametrical method, it is well suited as a data-analytic approach for situations in which a parametric model for intra-individual and inter-individual variation is available, for example, based on theory or previous empirical work. However, in many situations such a model might not be readily available. A typical example are brain imaging data such as EEG data. In this case, the conventional analysis approach treats both the intra-individual and the inter-individual variation as measurement error. In the second part of this thesis, I propose a new analysis strategy based on techniques

from the field of machine learning to account for inter-individual variation in EEG data sets by adopting a person-specific analysis approach (e.g., Molenaar & Campbell, 2009). The approach is general enough that it could also be applied to other brain-imaging methods such as magnetic resonance imaging (MRI).

In EEG studies, researchers are typically interested in estimating event-related potentials (ERPs). An ERP is the measured brain response caused by a given behavior, such as looking at a certain stimulus. To estimate person-specific ERPs, the behavior of interest is repeated multiple times for each person. The repetitions are commonly referred to as trials. The person-specific ERPs are obtained by averaging across all repetitions for each person. Thus, intra-individual variation in the the brain–behavior mapping is considered as measurement error.

To compare ERPs evoked by a given behavior between groups like younger and older adults, conventional statistical procedures like the t-test or ANOVA are commonly employed. Certain features of interest of the person-specific ERPs are used as the DV, e.g., the mean amplitude in a given time window. Group membership is the IV. Thus, while this approach specifically focuses on variation between groups, the inter-individual variation within each group is treated as measurement error. Consequently, conventional ERP analysis treats both inter-individual and intra-individual variation as measurement error. Accordingly, in order for the group-average ERPs to be representative of the true brain response for a given person at a given time point, both the intra-individual and inter-individual variation have to be sufficiently low. Empirical data suggests that at least the assumption of small inter-individual variation is not met for children and older adults, since an increased heterogeneity of functioning in behavioral tasks is observed (e.g., Astle & Scerif, 2011; Nagel et al., 2009; Werkle-Bergner, Freunberger, Sander, Lindenberger, & Klimesch, 2012).

One solution to account for the observed inter-individual variation is to adopt a person-specific analysis strategy as propagated, for example, by Molenaar (2013). Using the conventional ERP analysis approach, this could in principle be achieved by performing a separate statistical analysis for each person. However, the signal-to-noise ratio in EEG data is typically not high enough to make this approach feasible in practice. The statistical power of this approach would be too low for most available EEG data sets. Even worse, since the very idea of performing the analysis on the person-specific level means allowing the link between behavior and EEG signal to vary between persons, many statistical tests would have to be performed, one for each hypothesized univariate link between behavior and brain (i.e., for each feature of the EEG like the mean amplitude within a given channel and time window), decreasing statistical power even further.

To solve the problems of a low signal-to-noise ratio and low statistical power, I propose adapting techniques from the field of *brain–computer interface* (BCI) research (Wolpaw & Wolpaw, 2012). Here, the goal is to predict a participants' brain state given the data of only a single trial. Thus, one may see this data analysis method as a reversal of the conventional ERP approach. Instead of finding a mapping from a behavior to one feature of an ERP, and repeating this process for different features (a so-called mass univariate approach), one tries to find a mapping from the complete EEG signal

of one trial to the behavior (multivariate approach). In the MRI literature the former approach is known as *encoding* (from behavior to data) and the latter as *decoding* (from data to behavior). Another name for the decoding approach within the MRI community is *multi-voxel pattern analysis* (MVPA). Aggregating over time and space solves the two main problems of person-specific ERP analysis, as previously identified. First, the signal-to-noise ratio can be increased by aggregating. Second, only one statistical test has to be performed per person.

In conventional BCI studies the ultimate goal is to enable people to control a device just with their thoughts, more precisely through specific, reliable activity patterns of their brains. To evaluate the utility of such an approach, the accuracy of the estimated brain–behavior mapping is of interest. In this study, however, interpreting individual brain–behavior mapping as person-specific model is of primary interest. The techniques employed within BCI research as well as in this work stem from the field of machine learning. In that field, mappings from IVs to the DV are typically found by using algorithms that select the best mapping based on a set of candidate mappings and a data set. Rather than a theory-driven confirmatory approach, the BCI approach may be seen as exploratory and entails selection between a large set of competing hypotheses that are data-driven. In BCI, an interpretation of the selected mapping is not of interest, and arbitrary complex mappings can be used as candidate mappings as a result. In contrast, the space of candidate mappings needed to be carefully restricted to be interpretable but still powerful enough to achieve a satisfying accuracy in this work. A complex, neural network regression approach, for example might be powerful but not interpretable. A simple threshold model on a particular feature of the EEG signal (e.g., the mean amplitude within a given fixed time windows and channel) on the other hand might be interpretable but not powerful enough. To this end, I use a set of candidate models based on common spatial pattern (CSP) and linear discriminant analysis (LDA), which is commonly used for creating BCIs based on rhythmic neural activity (e.g., Blankertz et al., 2010; Fazli et al., 2009), and show how the resulting mapping can be meaningfully interpreted as person-specific models. In a second step, I demonstrate how the inter-individual variation of the found person-specific mappings can be explored as a proxy for the inter-individual variation of the true brain–behavior mapping.

To test the feasibility and utility of my approach, I re-analyzed data from a study that targeted brain oscillatory mechanisms for working memory (WM) selection and maintenance in a sample including children, younger, and older adults (Sander, Werkle-Bergner, & Lindenberger, 2012). We (Karch, Sander, von Oertzen, Brandmaier, & Werkle-Bergner, 2015) selected this data set because we expected true inter-individual variation in the brain–behavior mapping, particularly in the older adults. Mixed results on the link between EEG and behavior exist with regard to the mechanism of WM. Specifically, Sander et al. (2012) found evidence for attentional effects on the modulation of alpha power, whereas Vaden, Hutcheson, McCollum, Kentros, and Visscher (2012) did not observe any attention-related modulation in older adults. One explanation for the mixed results may be the increased inter-individual variation in older adults as compared to younger adults, which is ignored when using conventional analysis approaches but not

when applying the new analysis approach suggested here.

The results of using the proposed person-specific method to analyze the WM data set show that my approach leads to more accurate models than does the conventional analysis approach ignoring inter-individual variation. Thus, I challenge the implicit assumption that the observed inter-individual variation is measurement error. Interestingly, some aspects of the within-group inter-individual variation, as obtained by the proposed person-specific analysis strategy, was lower in older than in younger adults, which is contrary to what we (Karch et al., 2015) expected.

Thus, in this thesis, I present two new methods for the analysis of psychological data based on machine learning methods. The first, GPPM, is a new analysis method for panel data that both generalizes and extends conventional panel methods such as longitudinal SEM. The second method is a new analysis method for EEG data in which multivariate models are first estimated on the person level and then aggregated on the group level, explicitly addressing the within-group inter-individual variability, which is ignored by conventional analysis approaches.

1.3. Outline

In Chapter 2, I will start with a recapitulation of basic modeling principles that are necessary to understand the methods I will introduce later. The psychological curriculum rarely goes beyond frequentist statistics, which I will repeat nevertheless for the sake of completeness. Bayesian statistics and machine learning principles for modeling are usually left out. I will explain all three approaches from a single, unified perspective.

Before presenting the new longitudinal panel modeling method GPPM in chapter Chapter 3, I will introduce the techniques it is based on, the flexible Bayesian regression method GPR, and the time series analysis method GPTSM, which in turn is based on GPR. Because GPPM is later compared to the classical panel modeling method longitudinal SEM, and SEM was used as a source of inspiration for the development of GPPM, I also introduce SEM and, in particular, longitudinal SEM in Chapter 3.

I will discuss the advantages of GPPM over conventional panel methods, in particular longitudinal SEM and multiple-subject SSM, in Chapter 4, both theoretically and by providing examples. The examples also serve as a demonstration of GPPM. In this chapter, I furthermore compare the fitting speed of SEM and GPPM software, theoretically and empirically. This was done to investigate whether formulating models in their GPPM representation may speed up parameter estimation as compared to the SEM representation.

In Chapter 5, I will present the person-specific EEG analysis in which first person-specific models are obtained and then second these person-specific models are summarized on the group level. I will also present the validation of the method based on data from the study targeting brain oscillatory mechanisms for WM selection and maintenance.

In Chapter 6, I summarize and discuss the results and contributions of my thesis as well as future research directions.

2. Statistical Inference and Supervised Machine Learning

Statistical inference revolves around making probabilistic statements about the properties of a population based on a random, observed subset of the population. In psychology, an example for a property is the average general intelligence factor g of a given group, for example, older adults. The dominating approach to statistical inference in psychology is frequentist inference.

Supervised machine learning (abbreviated as supervised learning in the remainder) on the other hand, focuses on obtaining a prediction function on the base of a random, observed subset of the population. This function should predict a variable reliably based on some other variables for the complete population. Both statistical inference and supervised learning use a finite data set to obtain knowledge that should generalize to the population.

The popularity of Bayesian inference, in particular, (Lee & Wagenmakers, 2005; Wagenmakers, Wetzels, Borsboom, & van der Maas, 2011) but also of supervised learning (Markowetz, Błaszkiewicz, Montag, Switala, & Schlaepfer, 2014; Yarkoni, 2012) is growing within psychology. However, both are still not part of many psychological curricula. This chapter, thus, serves the purpose of introducing Bayesian inference and supervised learning as alternative methods for modeling empirical data and making inferences from random samples. Before introducing Bayesian inference, it seems useful to reintroduce frequentist inference in a more formal manner than typically done in psychology teaching. This paves the way for the presentation of frequentist inference, Bayesian inference, and supervised learning within a single, coherent framework.

I will start this chapter by introducing the foundations of statistical inference in Sections 2.1 and 2.2. A section explaining the frequentist and the Bayesian variants of statistical inference will follow. Then, I will introduce supervised learning and its connection to statistical inference. In the final section, I will compare the different approaches (Bayesian, frequentist, and supervised learning) to model selection.

2.1. Foundations of Statistical Inference

A huge number of good textbooks introduce probability theory (e.g., DeGroot & Schervish, 2011; Rice, 2007; Taboga, 2012b; Wasserman, 2004). Thus, I provide only a swift reminder to refamiliarize readers with probability theory in Appendix A. As the probability theory terminology differs slightly between authors, Appendix A also serves to present the terminology as used in this work.

The statistical literature aimed at psychologists typically introduces the topic of statistical inference in a rather informal manner (compare, for example, the current standard statistics book for psychologists in Germany [Eid, Gollwitzer, & Schmitt, 2013] with any general statistics text book [e.g., Wasserman, 2004]). This level does not suffice to understand the following chapters. Literature aimed at mathematicians or statisticians, however, requires more background knowledge than even the typical quantitative psychologist has learnt. Thus, I have tried to make this chapter as rigorous as possible without using advanced mathematical concepts (e.g., measurement theory and Borel sets) but limit myself to high school-level mathematics, that is, a good understanding of set theory, linear algebra, and calculus. I also assume that the reader has had first contact with frequentist inference.

The central task of statistical inference is to make probabilistic statements about the properties of a population based on a *sample*. Formally, a sample is defined as follows.

Definition 2.1.1. A *sample* y is a realization of a *random vector* Y.

Remark 2.1.2. A random vector is the generalization of a random variable. It is simply a vector that contains multiple random variables as entries. For a formal definition see Equation A.1.3 in Appendix A.1.

Example 2.1.3. For a longitudinal study during which 3 properties of 100 persons have been measured at 10 different time points the $3 \times 100 \times 10$ data cube is a sample.

Thus, the task of making probabilistic statements about the population is formalized as making statements about the distribution that generated the sample. For now, I will describe this distribution via its distribution function F_Y^* and call it the *true (generating) distribution* in the following.

The *statistical model* is a central concept of statistical inference. It encodes the assumptions about the true distribution F_Y^* before having observed the sample y. The assumptions are described by a set of candidate probability distributions for F_Y^*. Thus, a statistical model is merely a (typically infinite) set of probability distributions. For example, a statistical model is that the generating distribution F_Y^* is a Gaussian distribution. Formally, a statistical model can be described as follows.

Definition 2.1.4. Let y be a sample with the corresponding n-dimensional random vector Y. Let B be the set of all n-dimensional distribution functions:

$$B := \{F : \mathbb{R}^n \to \mathbb{R}^+ \text{such that } F \text{ is a distribution function}\}$$

A subset $\mathcal{M} \subset B$ is called a statistical model for Y.

Remark 2.1.5. As notation, I will often use

$$Y \sim \mathcal{M},$$

which reads that the statistical model for the random vector Y is $\mathcal{M}$, or in other words that the random vector is distributed according to one of the distributions within $\mathcal{M}$.

If the assumptions about the generating distribution F_Y^* are wrong, the statistical model is said to be misspecified; otherwise it is correctly specified.

Definition 2.1.6. Let F_Y^* be the true generating distribution function and $\mathcal{M}$ a corresponding statistical model. If $F_Y^* \in \mathcal{M}$, then the statistical model is said to be *correctly specified*; otherwise it is *misspecified*.

In other words: the true generating distribution F_Y^* has to be one of the probability distributions that constitute the statistical model.

Model specification is a crucial part of statistical inference. It refers to the process of developing a statistical model for an unknown distribution F_Y^*. It is important to realize that all traditional statistical inference procedures rely on the assumption that the statistical model is correctly specified. Otherwise, inference is formally invalid and to an unknown extent wrong.

In general, a statistical model $\mathcal{M}$ can be indexed by a possibly infinite dimensional index set Θ such that

$$\mathcal{M} = \{F_\theta : \theta \in \Theta\}.$$

In the following, I will refer to this index set Θ as the *parameter space* and to its elements $\theta \in \Theta$ as *parameter values*. If a statistical model is correctly specified, a parameter θ^* exists such that $F_{\theta^*} = F_Y^*$. Thus, in accordance with the literature, I will denote the generating distribution F_Y^* by θ^*.

This work is only concerned with statistical models for *continuous* random vectors Y, for which the statistical model can equivalently be expressed in terms of a set of *probability density functions (pdfs)* instead of a set of distribution functions. Therefore, in the remainder, statistical models will be described as a set of pdfs:

$$\mathcal{M} = \{p_\theta(y) : \theta \in \Theta\}.$$

In psychology, a sample y is typically a sample from a population of persons. That is, the sample has the form $y = [y_1, \ldots, y_N]^\top$, where N refers to the number of persons. Every y_i refers to the observed data for one person. The corresponding random vector Y_i, of which y_i is a realization, contains the variables of interest, for example, the general intelligence factor g and socioeconomic status for a cross-sectional data set or repeated measurements of general intelligence factor g and socioeconomic status for a panel data set.

Typically, the assumption is made that the observations y_i for the different persons do not influence each other, for example, if it is known that the general intelligence factor g for person 7 is 110, this will not alter the expectation regarding the general intelligence factor of any other (unrelated) person. Formally, this is represented by the assumption that the random vectors $\{Y_i : i \in 1, \ldots, N\}$ are *mutually independent*.

Another assumption that is commonly made is that the observed data y_i for each person are a realization of the same probability distribution. That is, the random vectors $\{Y_i : i \in 1, \ldots, N\}$ are not only independent; every random vector is distributed according to the same distribution. Taken together, these two assumptions are known as the *independent and identically distributed (iid)* assumption.

Definition 2.1.7. A set of random vectors $\{Y_i : i \in 1, \ldots, N\}$ is *independent and identically distributed (iid)* if, and only if, (iff) each random vector Y_i has the same probability distribution and if the random vectors are mutually independent.

If the iid assumption is made, the statistical model for the random vector Y can be determined by specifying a statistical model for the random vectors Y_i. Because the random vectors are identically distributed, it is reasonable to use the same statistical model for each random vector Y_i. The statistical model for the sample follows from the mutual independence assumption. Formally, if for all $i \in \{1, \ldots, N\}$ the statistical model for each random vector Y_i is

$$Y_i \sim \{p_\theta(y_i) : \theta \in \Theta\},$$

it follows that

$$Y \sim \left\{ \prod_{i=1}^{N} p_\theta(y_i) : \theta \in \Theta \right\}.$$

The two most common types of inference are *estimation* and *hypothesis testing*. In estimation, the aim is to eliminate some of the members of the statistical model as candidates for the true probability distribution θ^*. That is, in estimation, a restriction, which is formalized via a subset $\Theta_R \subset \Theta$ of the parameter space Θ, is selected such that a desirable property is fulfilled. For *point estimation*, the desired restriction implies a single point, that is, $|\Theta_R| = 1$. In point estimation it is common to denote the subset that implements the restriction via $\hat{\theta} \in \Theta$. In psychology, the most common point estimation technique is *maximum likelihood (ML)* estimation.

Example 2.1.8. Let Y consist of N iid random vectors Y_i and the statistical model for the random vectors Y_i be $Y_i \sim \{\mathcal{N}(\mu, 1) : \mu \in \mathbb{R}\}$, then the ML estimate $\hat{\mu}$ for the mean μ based on a sample $y = [y_1, \ldots, y_N]^\top$ is

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

If the restriction Θ_R can contain more than one point, that is $|\Theta_R| > 1$, one speaks of set estimation. Confidence intervals are a popular example for set estimation.

Example 2.1.9. Let the situation be set up as in Example 2.1.8, then a 95% confidence interval for the mean μ is

$$[\hat{\mu} - 1.96, \hat{\mu} + 1.96].$$

Hypothesis testing is strongly linked to set estimation. A restriction $\Theta_R \subset \Theta$ is proposed and either rejected or not.

Example 2.1.10. Let the situation be set up as in Example 2.1.8. A hypothesis test that rejects the restriction $\mu = 0$ if $|\hat{\mu}| > 1.96$ has the size $\alpha = 0.05$.

Frequentist and Bayesian inference provide all these central inference types but achieve them in quite a different manner. I will try to convey the differences in the following. However, I will first present the endeavor of statistical inference as a decision problem. This will provide the formal basis to delineate the difference between the Bayesian and the frequentist approaches.

I will use the following notation without explicitly repeating it in all subsequent theorems and definitions. Y is the random vector for which a statistical model $\mathcal{M} = \{p_\theta(y) : \theta \in \Theta\}$ is proposed, y denotes one realization of Y, i.e., a sample, Ω_Y is the support of the random vector Y, that is, the set of all possible values that the realizations y can obtain, i.e., $y \in \Omega_Y$.

2.2. Statistical Inference as a Decision Problem

All types of statistical inference can be interpreted as decision problems. Given a sample y, a decision to (a) reject a hypothesis or not, (b) for a set of likely parameters $\Theta_R \subset \Theta$, or (c) for a point estimate $\hat{\theta} \in \Theta$ has to be made. With *decision rules* and *actions*, statistical decision theory provides the necessary tools for a unifying treatment of these different types of statistical inference. For hypothesis testing, for example, the decision rule is a hypothesis test and the action is to either reject the hypothesis or not. More generally, a decision rule is defined as follows.

Definition 2.2.1. A (deterministic) *decision rule* is a function $\delta : \Omega_Y \to \mathscr{A}$ that maps every sample $y \in \Omega_y$ onto an action $\delta(y) = a \in \mathscr{A}$ (Berger, 1993, p. 9). The set of actions $\mathscr{A}$ can be any non-empty set.

Remark 2.2.2. For point estimation, the decision rule is an estimator (often denoted $\hat{\theta}(y)$) that maps every data set to one particular parameter $\hat{\theta}$; typically called an estimate. Thus, in this case the set of possible actions is the parameter space, $\mathscr{A} = \Theta$. For set estimation, the set of possible actions $\mathscr{A}$ is the set of all possible subsets of the parameter space, called power set and denoted 2^Θ. A set estimator maps every data set onto one member of the power set. For hypothesis testing, the set of possible actions is to either reject the proposed restriction $\Theta_R \subset \Theta$ of the parameter space or not. A hypothesis test maps every sample y to the decision to either reject a proposed restriction or not.

To evaluate a decision rule, its actions need to be evaluated. Each action is given a cost that depends on the true state of nature, i.e., the true generating probability distribution F_Y^* of Y. For the moment I ignore the fact that the true distribution F_Y^* is not known. The true distribution is assumed to be a member of the statistical model. As a consequence, the true distribution can be described by the parameter θ^*. By choosing a loss function, one can quantify how good an action is under the reality that is described by the true distribution θ^*.

Definition 2.2.3. Let $\mathscr{A}$ be a set of possible actions of a decision function $\delta : \Omega_Y \to \mathscr{A}$, then any non-negative function L defined on $\Theta \times \mathscr{A}$ is a loss function.

Example 2.2.4. For point estimation, the set of possible actions $\mathscr{A}$ corresponds to the parameter space. Thus, for point estimation the loss function is a function that maps the true parameter θ^* and a point estimate $\hat{\theta}$ to a cost. One example for a loss function used for point estimation is the squared loss: $L_{\text{sq}}(\theta^*, \hat{\theta}) = (\theta^* - \hat{\theta})^2$.

Remember that the loss function was introduced to evaluate decision rules. To do so, one can replace the action in the loss function by a decision rule δ that produces actions:

$$L(\theta^*, \delta(Y)). \tag{2.2.1}$$

However, Y is a random vector and, hence $L(\theta^*, \delta(Y))$ too. Even worse, the true state of nature θ^* is unknown. Therefore, there are two unknowns to get rid of before a decision rule δ can be evaluated. This is where Bayesian and frequentist inference disagree and branch off (Jordan, 2009). They solve this problem fundamentally differently, which is partly due to their different interpretations of the concept of probability.

Before I describe the respective frequentist and Bayesian solutions to the aforementioned problem, I want to recapitulate the three central forms of inference introduced at the beginning of this section in the language of statistical decision theory:

A point estimator is a decision function that maps any sample to one point within the parameter space, and therefore to a probability distribution.

Definition 2.2.5. A *point estimator* is any function $\delta : \Omega_Y \to \Theta$. For a specific sample y, $\delta(y)$ is called the point estimate.

A *set estimator* is a function that maps any sample to a subset of the parameter space, and hence to a set of probability distributions.

Definition 2.2.6. A set estimator is any function $\delta : \Omega_Y \to 2^{\Theta}$. For a specific sample y, $\delta(y)$ is called the set estimate.

The central task of hypothesis testing is to either reject a restriction, a subsection of the parameter space $\Theta_R \subset \Theta$, or not.

Definition 2.2.7. Let $\{\Theta_R, \Theta_R^{\complement}\} = \Theta$ be a partition of Θ. The hypothesis $H_0 : \theta^* \in \Theta_R$ is called the *null hypothesis*.

Definition 2.2.8. Let H_0 be a null hypothesis, then a (measurable) function

$\varphi : \Omega_Y \to \{0, 1\}$ is called a *statistical test* if it is used in the following way:

$$\varphi(y) = 1 \implies \text{reject the null hypothesis } H_0$$
$$\varphi(y) = 0 \implies \text{do not reject the null hypothesis } H_0$$

The set $R = \{y \in \Omega_Y : \varphi(y) = 1\}$ is called the *rejection region*. For the vast majority of statistical tests, the rejection region R can be expressed in the following form:

$$R = \{y : T(y) > c\},$$

where $T : \Omega_Y \to \mathbb{R}$ is called *test statistic* and $c \in \mathbb{R}$ *critical value*.

Definition 2.2.9. When performing a statistical test, two kinds or errors can occur:

- Rejecting the null hypothesis H_0 even though $\theta^* \in \Theta_R$. This is called a *type I error*.
- Not rejecting the null hypothesis H_0 even though $\theta^* \notin \Theta_R$. This is called a *type II error*.

2.3. Frequentist Inference

2.3.1. Foundations

This section is a generalization of the treatment in Wasserman (2004, Chapter 12). Frequentist inference is the most popular approach to statistical inference. The second most popular approach is Bayesian inference. As I have described in the previous section, a share of the differences between Bayesian and frequentist inference can be traced back to their different interpretations of probability.

In frequentist inference only probability measures belonging to random experiments that can (in theory) be repeated indefinitely are considered. Probability measures are defined via limiting frequencies of the outcomes of a corresponding random experiment.

Definition 2.3.1. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space (see Definition A.1.1 in Appendix A.1) with the sample space Ω, the Sigma-algebra $\mathcal{F}$, and the probability measure $\mathbb{P}$ corresponding to a random experiment that can be repeated indefinitely. Let n be the number of times that the experiment has been repeated so far. For a given event $A_i \in \mathcal{F}$, let n_i be the number of times the event A_i occurred after n repetitions of the experiment, then the *objective probability* of event A_i is defined as

$$\mathbb{P}(A_i) = \lim_{n \to \infty} \frac{n_i}{n}$$

A direct consequence of this is that frequentist inference interprets the true state of nature θ^* as a fixed unknown constant. Thus, frequentists do not allow themselves to impose a probability distribution on the parameter space Θ to express the uncertainty about the true state of nature θ^*, as is common in Bayesian inference.

The second central idea of frequentist inference is that the decision function should lead to good actions for all possible samples y. Therefore, the first step in frequentist inference is to take the expectation across all possible samples y to make the loss function independent of the currently observed sample. This expectation is called *frequentist risk*. Optimally, one would take this expectation with respect to the generating distribution θ^*. However, since the true distribution is not known, the frequentist risk of a decision rule depends on the assumed true distribution $\theta^* = \theta$. Thus, a frequentist risk for each value θ within the parameter space Θ of the statistical model exists.

Definition 2.3.2. The *frequentist risk* of a decision rule δ and parameter value θ is the expectation of a loss function over samples assuming that θ represents the true distribution:

$$R(\theta, \delta) = \mathbb{E}_\theta[L(\theta, \delta(Y))] = \int_{\Omega_Y} L(\theta, \delta(y)) p_\theta(y) \mathrm{d}y.$$

Note that the frequentist risk depends on the assumption θ about the true distribution in two places, (1) in the loss function and (2) in the pdf. One approach to make the frequentist risk independent of the assumption about the true distribution θ would be to try to find a decision rule that has the lowest risk (over all decision rules) for all possible true distributions, as represented by $\theta \in \Theta$. However, such a decision rule hardly ever exists. Consider the following example adapted from Wasserman (2004, p. 194):

Example 2.3.3. Let $Y \sim \mathcal{N}(\theta, 1)$ and the loss function be the squared loss $L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$. Two point estimators are compared: $\hat{\theta}_1(y) = y$ and $\hat{\theta}_2(y) = 1$. The frequentist risk for estimator $\hat{\theta}_1$ is $\mathbb{E}_\theta[(Y - \theta)^2] = 1$. The frequentist risk for estimator $\hat{\theta}_2$ is $(\theta - 1)^2$. Hence, for $0 < \theta < 2$ estimator θ_2 has less risk than estimator θ_1. For $\theta \in \{0, 2\}$ the risk of both estimators is 1. For $\theta < 0$ or $\theta > 2$, estimator θ_1 has less risk. Thus, neither of the estimators has less risk for all possible true states of nature $\theta \in \Theta$.

In the previous example, even the naive, constant estimator $\hat{\theta}_2$ had less risk than the much more sensible estimator $\hat{\theta}_1$ for some true states of nature θ. This motivates the need for different approaches to make the frequentist risk independent of the assumption that the true state of nature is θ.

One popular approach is the so-called *maximum risk*. This risk is defined as the risk of a decision rule in the worst possible scenario.

Definition 2.3.4. The maximum risk of a decision rule δ is

$$\bar{R}(\delta) := \sup_{\theta \in \Theta} R(\theta, \delta).$$

Loosely speaking, the maximum risk is calculated by iterating through all probability distributions as represented by θ, assuming that the true state of nature θ^* is θ, calculating the corresponding frequentist risk, and then using the least upper bound for all risks as maximum risk.

Note that the maximum risk is independent of both the sample y and the assumption about the true state of nature θ. Therefore, it provides a single number performance metric for a decision rule. Naturally, one would like to find the decision rule that minimizes this metric. If a decision rule has this property, it is called a *minimax* rule.

Definition 2.3.5. A decision rule δ^* is *minimax* iff

$$\bar{R}(\delta^*) = \inf_{\delta \in \Delta} \bar{R}(\delta),$$

where Δ is the set of all decision rules.

Thus a minimax rule, for a given statistical model, has the least maximum risk among all decision rules.

I will now introduce frequentist point estimation, set estimation, and hypothesis testing. In each section, I will first introduce them similarly as they are presented conventionally and will then link each technique to the decision theoretic perspective developed here.

2.3.2. Point Estimation

Remember, a point estimator is a function that maps any sample to one point within the parameter space, and consequently to a probability distribution.

In frequentist inference, ML estimators are typically used for parametric models.

Definition 2.3.6. Let $\mathcal{M} = \{p(y;\theta) : \theta \in \Theta\}$ be a parametric statistical model and y a sample. The likelihood function for parameter value θ is then defined as $L(\theta) = p(y;\theta)$. The ML estimate $\hat{\theta}$ is obtained by choosing $\hat{\theta}$ such that it maximizes the likelihood function. An estimator that maps every sample to the corresponding ML estimate is called a ML estimator.

The link between ML estimation and the decision theoretic perspective developed in the previous section is less pure than for set estimation and hypothesis testing. Starting out from the decision theoretic perspective, one would ideally use the minimax estimator based on a given loss function and statistical model for point estimation. However, finding minimax estimators is difficult for most common combinations of statistical model and loss function. In contrast, ML estimators are applicable to every parametric model. Luckily, "for parametric models that satisfy weak regularity conditions the maximum likelihood estimator is approximately minimax" (Wasserman, 2004, p. 201). For more information on the favorable properties of the maximum likelihood estimator, see Taboga (2012d).

ML estimation is not a frequentist method per se. It can also be derived from a Bayesian perspective. However, ML estimation is often used as the first step for frequentist inference procedures. It is required for many important frequentist hypothesis tests and confidence intervals, both of which, as I will show in the following sections, are inherently frequentist approaches. In SEM, for example, the ML estimate is required for the likelihood-ratio test as well as Wald- and likelihood-based confidence intervals (see Section 3.2.2).

2.3.3. Set Estimation

Remember that a set estimator is a function that maps any sample to a subset of the parameter space, and thus to a set of probability distributions.

The frequentist idea of set estimation is that an accurate estimate is produced for any sample y. How well a set estimator performs is quantified by the probability that its set estimates contain the true value θ^*. In the following definition and the remainder of this section, $\mathbb{P}_\theta$ denotes the probability measure implied by the probability distribution that is described by the parameter value θ.

Definition 2.3.7. Let θ^* be the true state of nature and δ a set estimator, then

$$\mathbb{P}_{\theta^*}(\theta^* \in \delta(Y))$$

is called the *coverage probability* of the set estimator δ.

The true state of nature θ^* is usually unknown. Therefore, the maximum risk idea is applied again. A set estimator is evaluated in terms of its coverage probability in the worst case. This is called *level of confidence*. That is, if a set estimator has a confidence level of $x\%$, this means that repeated application of it yields at least $x\%$ of the set estimates containing the true value.

Definition 2.3.8. Let δ be a set estimator. Its *level of confidence* is

$$\gamma = \inf_{\theta \in \Theta} \mathbb{P}_\theta(\theta \in \delta(Y)).$$

For a particular sample y, $\delta(y)$ is called the *confidence region*. If the confidence region is univariate, i.e., $\delta(y) \subset \mathbb{R}$, it is called the *confidence interval*.

Remark 2.3.9. Often the level of confidence is ascribed to the confidence region instead of the set estimator. It is for example common to speak of a 95% confidence region, which implies that there is a probability of 95% that the confidence region contains the true parameter. I do not think that this is a particularly wise choice of words. One particular confidence region either contains the true state of nature θ^* or not. The level of confidence is a property of the set estimator that generated the confidence region. I think that this choice of words has contributed to the fact that confidence sets are often

misunderstood.

I have not yet explained why confidence set estimators are a frequentist concept. The following theorem explicitly links confidence set estimators to the maximum risk principle developed in Section 2.3.1. The theorem was taken from Berger (1993, p. 22). The proof is my own work. For this theorem and the remainder of this section, $\mathbb{E}_\theta[f(Y)]$ denotes the expectation of a random variable $f(Y)$ given that the distribution for the random variable Y is as described by the parameter value θ. f may be any deterministic mapping.

Theorem 2.3.10. An equivalent definition of a set estimator with confidence level γ is a decision rule $\delta : \Omega_Y \to 2^\Theta$ with a maximum risk $\sup_{\theta\in\Theta} R(\theta, \delta) = 1 - \gamma$ under the loss function

$$L(\theta, \delta(y)) = \begin{cases} 0 & \text{if } \theta \in \delta(y) \\ 1 & \text{if } \theta \notin \delta(y) \end{cases}.$$

Proof:

$$\mathbb{P}_\theta(\theta \in \delta(Y)) = 1 - \mathbb{E}_\theta[L(\theta, \delta(Y)] = 1 - R(\theta, \delta)$$

Thus,

$$\inf_{\theta\in\Theta} \mathbb{P}_\theta(\theta \in \delta(Y)) = 1 - \sup_{\theta\in\Theta} R(\theta, \delta) = \gamma.$$

□

2.3.4. Hypothesis Testing

Remember that a hypothesis test is a function that maps any sample to the decision either to reject or not to reject the null hypothesis.

The main idea of frequentist hypothesis testing is again to make the correct decision for the majority of observed samples y. That is, the probability of making type I and type II errors should both be as small as possible.

To quantify the type I and type II error of a given statistic test the *power function* is employed.

Definition 2.3.11. The *power function* $\beta(\theta)$ of a statistical test φ expresses the probability of rejecting the null hypothesis under the assumption that the true distribution θ^* is θ

$$\beta(\theta) = \mathbb{P}_\theta(\varphi(Y) = 1) = \mathbb{E}_\theta\left[\varphi(Y)\right] = \int_{\Omega_Y} \varphi(Y) p_\theta(y) \mathrm{d}y$$

If the null hypothesis is true, that is, $\theta^* \in \Theta_R$, the power function $\beta(\theta^*)$ quantifies the probability of a type I error for a given statistical test. If the null hypothesis is false, that is, $\theta^* \notin \Theta_R$, the inverse of the power function $1 - \beta(\theta^*)$ quantifies the probability of a type II error. Thus, the value of the power function $\beta(\theta^*)$ of a perfect test would be 1 if $\theta^* \notin \Theta_R$ and 0 if $\theta^* \in \Theta_R$.

However, the true state of nature is not known. Therefore, one can only obtain upper bounds for the type I and the type II error. First, for the type I error:

Definition 2.3.12. The *size* of a statistical test is

$$\alpha = \sup_{\theta \in \Theta_R} \beta(\theta).$$

The size of a statistical test is a least upper bound for the type I error. For the type II error:

Definition 2.3.13. The *power* of a statistical test is

$$\beta = \inf_{\theta \in \Theta_R^{\complement}} \beta(\theta).$$

The power of a statistical test is the greatest lower bound for the probability of rejecting the null hypothesis if the null hypothesis is false. Thus, $1 - \beta$ is a least upper bound for a type II error.

Typically, there is a trade-off between the inverse of the power $1 - \beta$ and the size of a statistical test. Extending the rejection region R of a statistical test increases its power and size, whereas reducing the rejection region decreases its power and size.

Instead of fixing a rejection region corresponding to a given power and size, an alternative approach can be used. The alternative procedure is to start with a statistical test of size $\alpha = 1$ and decrease α until the null hypothesis is no longer rejected.

Definition 2.3.14. Let H_0 be a null hypothesis and φ_α a corresponding statistical test of size α, then

$$p = \inf\{\alpha : \varphi_\alpha(y) = 1\}$$

is called *p-value*.

For frequentist hypothesis tests I will establish the link to the decision theoretic perspective indirectly by showing that they are strongly connected to confidence set estimators. The following theorem shows how confidence set estimators can be used to construct hypothesis tests and vice versa.

Theorem 2.3.15.

1. Assume that for every point null hypothesis $\theta^* = \theta$ there is a statistical test φ_θ of size α, then the set estimator

   $$\delta(y) = \{\theta \in \Theta : \varphi_\theta(y) = 0\}$$

   has a confidence level of $1 - \alpha$.

2. Let δ be a set estimator with confidence level γ, then by using the rejection region

$$R_\theta = \{y : \theta \notin \delta(y)\}$$

a statistical test φ_θ with the size α of 1-γ can be obtained for every point null hypothesis $\theta^* = \theta$.

Proof: For every statistical test φ_θ the restriction that is tested is that $\theta^* = \theta$. Remember that the size of a statistical test α is defined as the maximum probability of rejecting the null hypothesis if it is true. If the null hypothesis only contains one distribution, it follows that for every statistical test the corresponding size is:

$$\alpha = \beta_{\varphi_\theta}(\theta),$$

where β_{φ_θ} denotes the power function of the corresponding test.

1. The confidence level of a set estimator $\delta(y)$ is

$$\inf_{\theta \in \Theta} \mathbb{P}_\theta(\theta \in \delta(Y)).$$

For every parameter value $\theta \in \Theta$, the corresponding hypothesis test φ_θ is used to decide whether the parameter value should be in this confidence set or not. Thus, the random set $\delta(Y)$ is $\{\theta : \varphi_\theta(Y) = 0\}$. For every parameter θ, the probability of it being in the confidence set is thus $\mathbb{P}_\theta(\varphi_\theta(Y) = 1)$; or equivalently $1 - \mathbb{P}_\theta(\varphi_\theta(Y) = 0)$. Hence, the confidence level can be reexpressed as

$$\inf_{\theta \in \Theta} (1 - \mathbb{P}_\theta(\varphi_\theta(Y) = 1)) = 1 - \sup_{\theta \in \Theta} \mathbb{P}_\theta(\varphi_\theta(Y) = 1).$$

$\sup_{\theta \in \Theta} \mathbb{P}_\theta(\varphi_\theta(Y) = 1)$ is the least upper bound for any test φ_θ to reject the null hypothesis $\theta^* = \theta$ if it is true. By assumption, this term is α for any hypothesis test φ_θ. Using the power function, this can be formally expressed as follows

$$\mathbb{P}_\theta(\varphi_\theta(Y) = 1) = \beta_{\varphi_\theta}(\theta) = \alpha.$$

It follows that $1 - \sup_{\theta \in \Theta} \mathbb{P}_\theta(\varphi_\theta(Y) = 1) = 1 - \alpha$, which concludes the proof of the first part.

2. The size α of a statistical test φ_θ for the hypothesis $\theta^* = \theta$ is

$$\alpha = \beta_{\varphi_\theta}(\theta).$$

By substituting the power function by its definition, one obtains

$$\beta_{\varphi_\theta}(\theta) = \mathbb{P}_\theta(\varphi_\theta(Y) = 1).$$

The probability of the hypothesis test φ_θ to reject the null hypothesis is equal to the probability of the sample being in the rejection region $\mathbb{P}_\theta(Y \in R_\theta)$. The

rejection region is built such that every sample creating a confidence set that does not include the null hypothesis parameter value θ is included. Thus, the probability of the sample being within the rejection region is equal to the probability of the null hypothesis parameter value θ not being in the confidence region $\mathbb{P}_\theta(\theta \notin \delta(Y))$. This in turn can be described by its complement $1 - \mathbb{P}_\theta(\theta \in \delta(Y))$. Since the confidence level of the confidence region estimator is γ, $\mathbb{P}_\theta(\theta \in \delta(Y)) \geq \gamma$. Thus, $1 - \mathbb{P}_\theta(\theta \in \delta(Y)) \leq 1 - \gamma$

□

Remark 2.3.16. The direction of constructing set estimators from statistical tests will be of particular importance in the remainder of this work. For a unidimensional parameter space, Theorem 2.3.15 shows how to create a confidence interval for the parameter θ. For a multidimensional parameter space, the application of Theorem 2.3.15 leads to a confidence region, but it is often of interest to construct a confidence interval for one parameter θ_p. This can be achieved in analogy to Theorem 2.3.15 by using hypothesis tests for the null hypothesis $\theta_p^* = \theta_p$.

2.4. Bayesian Inference

Bayesian inference also starts with the problem that for the evaluation of a decision function δ through equation

$$L(\theta^*, \delta(Y)) \tag{2.4.1}$$

the true parameter θ^* is unknown and Y is a random variable.

As I have shown in the last section, the defining characteristics of frequentist inference are to solve this problem by taking the expectation across samples and not making any distributional assumptions about the value θ of the true parameter θ^*.

The Bayesian solution is somehow the opposite. Instead of taking the expectation across samples, all inferences are conditioned on the actual observed data set. Additionally, distributional assumptions about the value θ of the true parameter θ^* are made. Bayesians are allowed to impose a distribution over model parameters because they are more liberal in their interpretation of the concept of probability. For a Bayesian, a probability measure is simply a function that describes the degrees of beliefs for the different events. Therefore, a pdf $p(\theta)$ can be imposed over the different parameter values θ. $p(\theta)$ is commonly called the *prior distribution*. In Bayesian inference the prior distribution is part of the statistical model.

Having observed a sample and making all inferences conditional on this, the prior distribution can be updated to a *posterior distribution* using Bayes' rule (Theorem A.2.10 in Appendix A.1). To do this, note that if it is assumed that θ is a realization of a random vector, the pdf describing the statistical model $p(y;\theta)$ is a conditional pdf $p(y|\theta)$. Thus, the pdf describing the posterior distribution over the parameters θ is

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int_\Theta p(y|\theta)p(\theta)}. \tag{2.4.2}$$

Many proponents of Bayesian inference typically report the posterior distribution as end result of statistical inference. However, set and point estimation, and hypothesis testing are also used. Bayesian inference procedures can be considered decision functions in equivalence to their frequentist counterparts. However, the *Bayesian expected loss* is used to evaluate them rather then the frequentist risk.

Definition 2.4.1. Let the pdf $p(\theta)$ describe the prior distribution of the parameters θ, $L(\theta, a)$ be a loss function, and y the observed sample, then

$$R_B(\delta, y, p) = \mathbb{E}_{p(\theta|y)}[L(\theta, \delta(y))] = \int_\Theta L(\theta, \delta(y))p(\theta|y) \tag{2.4.3}$$

is the *Bayesian expected loss* for the decision rule δ, given the prior $p(\theta)$ having observed the sample y.

The frequentist risk is an expectation across samples whereas the Bayesian expected loss is an expectation across parameters. The Bayesian expected loss can be directly calculated as it does not directly depend on the true state of nature θ^*. Since the posterior distribution $p(\theta|y)$ contains all information about the belief about θ^*, all statistical procedures only describe certain features of the posterior and are thus easy to derive.

2.4.1. Point Estimation

One popular Bayesian point estimator is the mean of the posterior distribution, that is,

$$\delta(y) = \int_{\theta\in\Theta} \theta p(\theta|y)\mathrm{d}\theta.$$

Since the full posterior is typically known, the mean estimator can be directly calculated. The mean estimator is the point estimator that minimizes the Bayesian expected loss with respect to the squared loss.

Another popular Bayesian point estimator is the maximum a posteriori (MAP) estimator. It is simply the mode of the posterior distribution. If the prior is flat, that is, $p(\theta) = c$ for some constant c, the MAP estimator is equivalent to the ML estimator.

2.4.2. Set Estimation

Credibility regions are the Bayesian counterparts to confidence regions. In contrast to confidence regions, credibility regions are easy to conceptualize.

Definition 2.4.2. A so-called $x\%$ *credibility region* $\Theta_R \subset \Theta$ is a subset of the parameter space that contains $x\%$ of the probability mass of the posterior distribution, that is, a subset Θ_R of the form such that

$$\int_{\Theta_R} p(\theta|y)\mathrm{d}\theta \geq x\%.$$

2.4.3. Hypothesis Testing

Bayesian hypothesis testing is more evolved than point estimation and set estimation. First, note that a null hypothesis can also be interpreted as a statistical model. Treating the parameters as random variables allows calculation of the probability (or density) of the data given the model, which is

$$p(y|m) = \int_{\Theta} p(y|\theta, m)p(\theta|m)\mathrm{d}\theta,$$

where m is the realization of a random variable M that encodes the choice of the statistical model as well as the prior. The probability of the model given the data is thus

$$p(m|y) = \frac{p(y|m)p(m)}{p(y)},$$

where $p(m)$ is the prior probability of the model. The probability of the data $p(y)$ is hard to compute. However, when comparing two models m_1,m_2, as done in hypothesis testing, one is only interested in the ratio of the probabilities

$$\frac{p(y|m_1)p(m_1)}{p(y|m_2)p(m_2)},$$

for which $p(y)$ disappears. The additional assumption that the prior probability $p(m)$ for every model is the same further simplifies this term to

$$K = \frac{p(y|m_1)}{p(y|m_2)}.$$

This term is known as the *Bayes factor*. If the ratio $K > 1$, the data support model m_1 over m_2. The greater K is, the stronger is the support for m_1.

2.5. Supervised Machine Learning

Ultimately, the main task of statistical inference is to reduce the uncertainty about the true generating distribution θ^* using a sample. Supervised machine learning is similarly concerned with finding a prediction function based on a finite data set that generalizes beyond the observed data set. I will first describe supervised machine learning without making the connection to statistical learning explicit as is typically done. For an elaborate introduction to machine learning, see e.g., Bishop (2006), Duda, Hart, and Stork (2001), Hastie, Tibshirani, and Friedman (2001), Murphy (2012).

The task is to find a function $f(x)$ based on IVs contained in the vector x (e.g., a pixel image of a digit) that predicts a DV y (e.g., the corresponding digit). Every pair (x, y) is assumed to be generated by an underlying probability distribution $p(x, y)$. Thus, every pair (x, y) can be interpreted as a sample. Additionally, it is typically assumed that all samples are independent of each other. The difference between unsupervised learning,

which I will not cover here, and supervised learning is that for unsupervised learning no observations of the DV y are available.

Learning algorithms are employed in order to obtain a prediction function. As input, a learning algorithms needs multiple examples (x_i, y_i), which are commonly gathered in a data set $D = \{(x_i, y_i) : i \in 1, \ldots, N\}$.

Definition 2.5.1. A learning algorithm $I(D)$ is a function that maps a data set D onto a prediction function $f(x)$. The data set D used as input for a learning algorithm is called the training set.

After a prediction function $f(x)$ has been obtained, a central question is how well it will perform on new examples that were not part of the training set. The first step to answering this question is to quantify what good performance means. This is done via a loss function, which assigns a cost to every prediction $f(x)$ based on the true value y.

Note that while both the loss functions used in statistical inference and supervised machine learning match the general definition (see Definition 2.2.3), they are used differently. For statistical inference, the true state of nature is represented by the true generating distribution, which is unknown. For supervised machine learning, the true state of nature is the true value y, which is directly observed and thus known. For statistical inference the action corresponds to the results of a statistical inference procedure (e.g., a point estimate). For supervised learning the action is the prediction $f(x)$.

For regression tasks, a popular loss function is again the squared loss

$$L(y, f(x)) = (y - f(x))^2.$$

Given a loss function, the performance of a prediction function $f(x)$ is simply the expected loss, which is again called risk.

Definition 2.5.2. The risk of a prediction function is

$$R(f) = \mathbb{E}_p[L(y, f(x))] = \int\int_{\Omega_X \times \Omega_Y} L(y, f(x))p(x, y)\mathrm{d}x\mathrm{d}y.$$

The risk depends on the unknown generating distribution $p(x, y)$. To estimate the risk, a finite data set D_h, the so-called test set, data set is used. To obtain an unbiased estimate, the test set D_h must not contain any sample that was included in the training set D_t.

Definition 2.5.3. The empirical risk of a prediction function based on a data set D, $|D| = N$ is

$$R(f) = \frac{1}{N} \sum_{(x,y)\in D} L(y, f(x)).$$

In practice, the training set and the test set are typically obtained by splitting a data set D. Thus, every member (x, y) of the data set is used for either training or testing.

Cross-validation can be employed to use every member (x, y) for training and testing. This method basically repeats the splitting into training and test set a number of times. The data set D is partitioned into n disjunctive subsets. The number of subsets n is commonly referred to as folds. For each subset D_i, the learning algorithm gets the remainder of the data $D \setminus D_i$ as input. The loss of all examples in D_i is calculated, and the overall estimate for the expected loss is simply the mean of the loss of all examples. Formally, the cross-validation risk is as follows.

Definition 2.5.4. Given a learning algorithm I, a data set D of size N, a loss function L, and a partition P of the data set D, the cross-validation risk of a learning algorithm is

$$R(I) = \frac{1}{N} \sum_{D_i \in P} \sum_{(x,y) \in D_i} L(y, I(D \setminus D_i)(x)).$$

In contrast to the empirical risk, the cross-validation risk is not a property of a prediction function but rather of a learning algorithm. It provides an estimate for the to-be-expected risk of the prediction function $f(x)$ that results from training a learning algorithm with a training set of the size $|D \setminus D_i|$. I speak of the set size despite the fact that $|D \setminus D_i|$ might be different for every D_i, because in practice the partition P is chosen such that $|D \setminus D_i|$ is roughly the same for every D_i.

2.6. Connections Between Supervised Learning and Statistical Inference

An obvious question is why new methods were developed for supervised learning instead of simply using the established statistical inference procedures. I will try to answer this question by describing the statistical inference solution to the supervised learning problem.

First, instead of using a learning algorithm $I(D)$ to obtain the prediction function, a statistical modeler could start with specifying a statistical model for the conditional distribution of dependent variable given the independent variables $p(y|x; \theta)$. Given a training set D, which in their terminology is a sample with many iid observations, they perform statistical inference. A frequentist might compute a point estimate for the parameter θ and use this to get a point estimate for the expected value of the DV Y, given some observed IVs x $\mathbb{E}(Y|x)$. Alternatively, a frequentist could compute confidence regions for the parameter θ and then use them to compute a confidence interval for $\mathbb{E}(Y|x)$. A Bayesian could obtain a posterior distribution $p(\theta|D)$ and link this with the likelihood function $p(y|x, \theta)$ to obtain a posterior predictive distribution $p(y|x, D)$ for the DV Y given the IVs x.

The problem with applying a purely statistical approach to solve the supervised learn-

ing problem is the core assumption of statistical inference, namely that the true generating distribution is a member of the statistical model. This assumption itself can be tested. However these tests typically lack power (Breiman, 2001, Section 5.2). Besides, for statistical inferences to be valid the correctly specified assumption needs to be guaranteed to be correct. This level of certainty can not be reached with any procedure that depends on data.

For typical supervised learning problems, making inferences that rely on the correctly specified assumption would be even more problematic than for conventional statistical inference problems. Statistical inference was originally developed for relatively simple problems, for which arguments that a statistical model is correctly specified might exist. Supervised learning on the other hand has been dealing with problems where the number of IVs is typically in the thousands, if not greater. Here, it becomes unrealistic to specify a set containing the true generating distribution. The prediction intervals obtained based on a misspecified statistical model, using either a frequentist or a Bayesian approach, are essentially meaningless. One remedy could be to use very broad statistical models. This however is accompanied by the problem that it would require huge data sets to obtain reasonable prediction intervals.

Supervised learning solves this issue by treating the learning algorithm as a black box. Inferences are only made about the outcome of the learning algorithm, the prediction function. The learning algorithm may be a statistical model in combination with an estimator. Indeed, many learning algorithms were developed using statistical inference ideas. However, other learning algorithms are motivated completely differently.

To estimate the risk without making any assumptions about either the learning algorithm or the true generating distribution, the risk is estimated nonparametically based on a holdout set or via cross-validation.

Reversing the question whether statistical inference can be used for supervised learning, the advantages of using supervised learning methods for typical statistical inference tasks will be addressed in the context of EEG analysis in Chapter 5.

2.7. Model Validation and Model Selection

2.7.1. Model Validation

I have emphasized that statistical inference, be it Bayesian or frequentist, relies on the assumption that the employed statistical model is correctly specified. Model validation in the context of statistical inference refers to the process of assessing this assumption. For a family of statistical models such as the GLM, there are typically a number of recommended approaches to assess model validity. For the GLM, for example, residual analysis is one approach to assess model specification. An alternative to model validation is sensitivity analysis (Saltelli, Chan, & Scott, 2000).

One general frequentist approach for model validation, which is used for many families of statistical models, is hypothesis testing. The rationale for doing this is to use the statistical model as the null hypothesis within the encompassing model of “all probability

distributions." If a null hypothesis test at a given size α does not reject the statistical model, the model is assumed to be correctly specified. The main problem with this approach is that it is an improper use of hypothesis tests as it accepts a non-rejection of the null hypothesis and is in contradiction to the underlying decision theory. In other words, we can never confirm the veracity of a model and can only fail to do so (Tomarken & Waller, 2003).

In Bayesian inference, the statistical model as well as the accompanying prior have to be validated. There are numerous approaches to do this. For a review, see Gelman et al. (2013, Chapter 6). One particular popularly approach involves comparing data generated by the posterior predictive distribution with actual observed data. For more details about this, see Gelman et al. (2013, Chapter 6).

In supervised learning, model validation consists of estimating the empirical risk of the prediction function. Thus, whereas the goal of model validation for frequentist inference and Bayesian inference is ultimately the same, it is very different for supervised machine learning.

2.7.2. Model Selection

An issue closely related to model validation is model selection. The task here is to choose the best model for a given data set. In practice, a set of candidate models is proposed and the best model is selected based on a model-scoring method. A model-scoring method typically trades off model fit against the parsimony of the model. A vast amount of different model-scoring methods exist. For some of the most popular, see Claeskens and Hjort (2008) and Burnham (2013).

One particular popular model selection procedure within frequentist inference is again a method based on the hypothesis test. Hypothesis testing for model selection works much like hypothesis testing for model validation. It can only be used to decide between two models. However, by repeating the process a decision can be taken between many models, but it only works if the models are nested within each other. That is, one model (called the restricted model) must be interpretable as the null hypothesis within the other model (called the full model). If the corresponding statistical test rejects the null hypothesis, the full model is chosen. Otherwise, the reduced one is selected.

For Bayesian inference, essentially the same strategy as the one for Bayesian hypothesis testing (Section 2.4.3) is commonly used: For every model m, the probability of the data under this model $p(y|m)$ is computed. The one that results in the highest probability of the data is selected. The models do not have to be nested.

For supervised learning, model selection corresponds to prediction function selection. The prediction function with the least empirical risk is commonly selected.

3. Gaussian Process Panel Modeling

In this chapter, I present the new panel modeling technique *Gaussian process panel modeling (GPPM)*. GPPM is a powerful technique that includes most conventional panel modeling techniques as special cases. However, it is also able to represent new models that cannot be represented using available methods. In addition, it provides a new perspective on known models, since its language for model specification is different.

GPPM is based on the flexible function-fitting approach *Gaussian process regression (GPR)*, which has recently gained popularity in the field of Bayesian supervised machine learning (Rasmussen, 2006). GPR is originally a Bayesian method. As a consequence, adapting the inference procedures used in GPR for GPPM also leads to Bayesian procedures. However, I have also developed frequentist inference procedures for GPPM. To do this, I borrowed ideas from SEM.

Before I introduce SEM, I will introduce the GLM in Section 3.1, to prepare the introduction of SEM and GPR. Both GPR and SEM can be interpreted as extensions of the GLM. I then focus on SEM and its accompanying frequentist inferences procedures in Section 3.2. Knowledge of SEM is also important for the comparison of longitudinal SEM and GPPM presented in the next chapter. In Section 3.3, I turn to GPR. In the following section, I will take the first step towards adapting GPR for the analysis of panel data by reviewing how GPR is used for time series analysis, which I call *Gaussian process time series modeling (GPTSM)*.

In Section 3.5, I will extend the time series modeling method GPTSM to the panel data method GPPM, whose favorable properties I will delineate in Chapter 4. A time series can be regarded as a special case of a panel data set for which only one person has been observed. Thus, to extend a time series modeling method to a panel modeling technique a mechanism to specify a simultaneous model for multiple persons has to be developed. My proposal for extending GPTSM constitutes the core of Section 3.5.

Since it is a time series modeling method, the focus of GPTSM is on modeling intra-individual variation. GPTSM allows specification of a wide range of models for intra-individual variation that exceed conventional time series methods such as autoregressive–moving average models by far. In a panel modeling method like GPPM, modeling the inter-individual variation is also crucially important. In Section 3.6, I will describe the possibilities to model inter-individual variation in GPPM.

Bayesian inference techniques, as derived from GPR, can be used as a statistical inference framework for GPPMs. However, frequentist inference is still the de-facto standard approach within psychology. Additionally, most existing panel modeling techniques are commonly employed in conjunction with frequentist inference. I will present a ML estimator, a frequentist hypothesis test procedure, and a confidence set estimator based on

the hypothesis test procedure for GPPM in Section 3.7. I will also provide recommendations for model selection and validation.

In Section 3.8, I will provide a brief overview of how I have implemented GPPM before closing this chapter with a comparison to related work.

3.1. General Linear Model

Both GPR and SEM can be interpreted as extensions of the GLM. The GLM is one of the most widely used statistical modeling techniques. While the term strictly only describes a family of statistical models, it is commonly used in conjunction with ML estimation and frequentist inference. The GLM is extensively used in statistics as well as in machine learning. In machine learning, it is commonly known as multiple linear regression (MLR). Within statistics, it is known under both names, MLR and GLM.

The GLM in conjunction with its ML estimator can be interpreted as a learning algorithm that is motivated by statistical inference principles. Statistical models of the conditional distribution of the DV Y_i given an observation for the IVs x_i that can be represented as GLM are of the form

$$Y_i|x_i \sim \{\mathcal{N}(x_i^\top \beta, \sigma_\epsilon^2) : (\beta, \sigma_\epsilon^2) \in \mathbb{R}^P \times \mathbb{R}_0^+\}.$$

The vector β is known as the weight vector in supervised learning. In psychology, the entries of β are typically referred to as regression coefficients. P denotes the number of IVs and consequently the dimensionality of the vector β. σ_ϵ^2 is commonly known as error variance. For GLM, the iid assumption is employed. It follows that the statistical model for a data set $D = \{(x_i, y_i) : i \in 1, \ldots, N\}$ of size N is

$$Y|X \sim \{\mathcal{N}(X\beta, \sigma_\epsilon^2 I_N) : (\beta, \sigma_\epsilon^2) \in \mathbb{R}^P \times \mathbb{R}_0^+\}.$$

The matrix

$$X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{bmatrix}$$

gathers the observed values. The random vector $Y = [Y_1, \ldots, Y_N]^\top$ represents the N observations of the DV. Thus, $\mathbf{y} = [y_1, \ldots, y_N]^\top$ is a realization of the random vector Y. I_N is the identity matrix of size $N \times N$.

When the GLM is used for statistical inference, the goal is to make probabilistic statements about the values of the parameters $(\beta, \sigma_\epsilon^2)$. Most often the statements of interest are about the regression coefficients within β, and not about the error variance. Most parametric tests as employed in psychological research (like the t-test and ANOVA) employ a variant of the GLM as the statistical model (Cohen, Cohen, West, & Aiken, 2003). They usually differ in the employed design matrix X and the probabilistic statements that are made about the coefficients within β. For inference, frequentist concepts (ML estimator, frequentist confidence regions, and hypothesis tests) are commonly used. For a detailed treatment of this topic, see, e.g., Cohen (1968, 6, Pt.1).

When the GLM is used for supervised learning, its statistical model is employed as the basis of a learning algorithm. Recall that a learning algorithm returns a prediction function $f(x)$ based on a data set D.

Here, it is worth commenting on slight notational differences between the GLM as used in psychology and the GLM as used in supervised learning. In supervised learning, to emphasize that the prediction function should produce good predictions for any value of the IV, one particular value of the IVs is denoted by x. In psychology, one particular value of the IVs is typically denoted as x_i to emphasize that it is typically part of an observed data set and refers to the IVs of one person. Thus, in the remainder of this section $Y|x$ equates to $Y_i|x_i$ in the notation of the GLM introduced at the beginning of this section.

I have already presented one approach to turn a statistical model on a conditional distribution into a learning algorithm. Specifically, a point estimate $\hat{\theta}$ for the statistical model is first obtained. The point estimate represents a particular conditional distribution. The expected value of the DV given the IVs under this conditional distribution $\hat{\theta}$ is used as prediction function, that is, $f(x) = \mathbb{E}_{\hat{\theta}}[Y|x]$. For the GLM, any parameter estimate $\hat{\theta}$ represents a different Gaussian distribution for the conditional distribution of $Y|x$. Since the mean of a Gaussian random vector is independent of its variance, $\mathbb{E}_{\hat{\theta}}[Y|x]$ is independent of the value of the error variance σ_ϵ^2. Thus, it suffices to obtain a point estimate for the weights β. Given a data set $D = (X, \mathbf{y})$, the ML estimate for the weights β is

$$\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}.$$

Proofs of this can be found in, e.g., Bishop (2006, Section 3.1.1). Using the ML estimate $\hat{\beta}$, the prediction function becomes $f(x) = x^\top \hat{\beta}$. This prediction function can then be evaluated, for example, by measures of explained variance (which is proportional to the squared loss). To obtain an estimate, either the hold-out method or cross-validation is typically employed. For a detailed treatment of MLR as used in supervised learning, see, e.g., Bishop (2006, Chapter 3).

3.2. Structural Equation Modeling

For the development of frequentist inference procedures for GPPM, I borrowed central ideas from SEM. I will later also compare GPPM with longitudinal SEM. Thus, providing an introduction to SEM seems appropriate.

SEM is a statistical method widely used within psychology. SEM allows the joint analysis of latent and observed variables and their interrelations, while explicitly accounting for measurement error. Importantly, SEM provides an unified and comprehensive statistical approach to test hypotheses on these interrelations, and allows for complex inferences on multivariate, correlational data.

A large body of work on the topic of SEM exists. The conventional reference remains Bollen (1989). Kaplan (2009) covers more recent developments. The most exhaustive

treatment is Hoyle (2014). For the mathematically less inclined readers, Kline (2011) provides a good introduction. SEM is virtually unused in machine learning. It is traditionally used in conjunction with frequentist inference. Of the different equivalent notations for SEM, I have selected the reticular action model (RAM) notation (McArdle & McDonald, 1984).

3.2.1. Structural Equation Models

The SEM can be interpreted as a generalization of the GLM. In the GLM, using the notation common in psychology, the DV Y_i is modeled as a linear function of the IVs x_i plus a random perturbation, which usually is thought to represent measurement error but subsumes any kind of unsystematic perturbation of the system of equations

$$Y_i = x_i^\top \beta + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2). \tag{3.2.1}$$

In SEM, this concept is extended to a system of linear equations between a set of variables. All variables are contained within the random vector V. In contrast to GLM, in texts about SEM, the person index i is usually omitted for the random vector V_i. Thus, I will also omit it here, although the logic of GLM notation would recommend writing V_i. A system of noisy linear equations among the variables represented in the vector V can be expressed as follows:

$$V = AV + U \tag{3.2.2}$$

The A matrix contains all linear equations between the variables, and the random vector U represents the random perturbations for the equations. The random vector U is assumed to be distributed according to a Gaussian distribution, i.e., $U \sim \mathcal{N}(m, S)$. The covariance matrix S is allowed to be any valid covariance matrix, i.e., any symmetric, positive semidefinite matrix. Thus, the random perturbations for the different equations can be assumed to be correlated.

In order to obtain an equation that has the variables V on only one side of the equality sign, Equation 3.2.2 can be rearranged as follows:

$$\begin{aligned} & V = AV + U \\ \iff & V - AV = U \\ \iff & (I - A)V = U \\ \iff & V = (I - A)^{-1}U \end{aligned} \tag{3.2.3}$$

I is an appropriately sized identity matrix.

For known and fixed matrix A, mean vector m, and covariance matrix S, $(I - A)^{-1}$ is a matrix and U a Gaussian random vector. Thus, using the linear transformation of Gaussian random vectors Theorem (Theorem A.3.8 in Appendix A.3), the distribution for variables within the random vector V implied by Equation 3.2.3 is

$$V \sim \mathcal{N}\left((I - A)^{-1}m, (I - A)^{-1}S(I - A)^{-1^\top}\right). \tag{3.2.4}$$

SEM uses the concept of latent variables. These are variables within V that are not directly observable. Thus, V may be partitioned in observable variables within the random vector Y and latent variables within the random vector L: $V = [Y^\top, L^\top]^\top$.

The implied distribution for the random vector Y, which represents the observed variables, is obtained by marginalizing over the joint distribution displayed in Equation 3.2.4. Marginalization can be expressed by a multiplication with a filter matrix F, such that $Y \sim FV$. The filter matrix F is a matrix of ones and zeros of the following form. Let H be the number of observed variables and J the number of latent variables, then $F = [I_H, 0_{HJ}]$, where I_H is the $H \times H$ identity matrix and 0_{HJ} the $H \times J$ matrix of zeros. Using the linear transformation of Gaussian random vectors Theorem (Theorem A.3.8 in Appendix A.3) again, the implied distribution for Y is

$$Y \sim \mathcal{N}\left(F(I-A)^{-1}m, F(I-A)^{-1}S(I-A)^{-1^\top}F^\top\right). \tag{3.2.5}$$

This describes one particular distribution for the observed variables within Y.

In order to obtain a statistical model, the matrices A, m, and S are parameterized. This leads to the following statistical model:

$$Y \sim \left\{\mathcal{N}\left(F(I-A(\theta))^{-1}m(\theta), F(I-A(\theta))^{-1}S(\theta)(I-A(\theta))^{-1^\top}F^\top\right) : \theta \in \Theta\right\}.$$

In SEM, every entry of A, m, S may be replaced by a single free parameter. Alternatively, the entry may be set to a fixed value. As a core assumption of SEM, for every parameter value $\theta \in \Theta$, the resulting distribution has to be Gaussian. This condition can be equivalently expressed as:

1. $S(\theta)$ is a covariance matrix, that is, a symmetrical positive semidefinite matrix.
2. $m(\theta)$ is a valid mean vector, that is $m(\theta) \in \mathbb{R}^{J+H}$.
3. $(I - A(\theta))$ is invertible.

Taken together the formal definition of a SEM is as follows.

Definition 3.2.1. A structural equation model (SEM) is a statistical model of the form

$$Y_i \sim \left\{\mathcal{N}\left(F(I-A(\theta))^{-1}m(\theta), F(I-A(\theta))^{-1}S(\theta)(I-A(\theta))^{-1^\top}F^\top\right) : \theta \in \Theta\right\},$$

where every entry $a_{ij}(\theta)$ of $A(\theta)$, every entry $m_i(\theta)$ of $m(\theta)$, and every entry $s_{ij}(\theta)$ of $S(\theta)$ is either a constant or corresponds to exactly one entry θ_p of the parameter vector θ. Additionally, for every parameter value $\theta \in \Theta$, $S(\theta)$ has to be a symmetrical positive semidefinite matrix, $m(\theta)$ has to be in $\mathbb{R}^{J+H}$, and the matrix $I - A(\theta)$ has to be invertible.

Under the iid assumption, the statistical model for a data set $\mathbf{y} = \{y_i : i \in 1, \ldots, N\}$ of multiple realizations y_i of the observable variables Y follows directly. Let $\mu(\theta) = F(I-A(\theta))^{-1}m(\theta)$ be the function describing the model-implied mean given a parameter

value θ, and let $\Sigma(\theta) = F(I - A(\theta)^{-1}S(\theta)(I - A(\theta))^{-1^\top}F^\top$ be the corresponding function describing the model-implied covariance. The statistical model for the data set y is then

$$Y \sim \left\{ \prod_{i=1}^{n} \mathcal{N}\left(y_i; \mu(\theta), \Sigma(\theta)\right) : \theta \in \Theta \right\},$$

where Y now is the random vector of which the N observations $\mathbf{y}$ are a realization. In psychology, every observation y_i typically corresponds to the data of one person.

Instead of specifying a SEM algebraically by a parameterization of the A, S, m matrices, it can also be graphically represented using a path diagram. Every path diagram corresponds to a certain algebraic representation and vice versa. For an introduction to path diagrams as used for SEM, see, e.g., Kline (2011).

For some applications it is important to be able to individualize the model for each person. For example, the LGCM requires individualization to allow the measurement time points to vary between persons. This motivates an extension of standard SEM, called definition variables.

Definition variables allow personalization of the matrices A,S,F, such that the fixed values are potentially different for each person. However, the entries of the matrices that contain free parameters are the same across all persons. Formally, the parameterization of the SEM matrices changes as exemplified for the A matrix in the following. Instead of only being parameterized by the parameters, the A matrix also depends on the definition variables values x_i for every person i: $A(\theta; x_i)$. As a result, the model-implied mean and covariance matrices additionally depend on the definition variables. Thus, formally, definition variables change the statistical model for a data set D to

$$Y \sim \left\{ \prod_{i=1}^{N} \mathcal{N}(y_i, \mu(\theta; x_i), \Sigma(\theta; x_i)) : \theta \in \Theta \right\}.$$

Another extension of standard SEMs is to drop the assumption that every entry of $A(\theta), S(\theta)$, and $m(\theta)$ is either a fixed value or a parameter θ_p. Instead of a single parameter, arbitrary functions of one or multiple parameters are allowed. Combining this idea with definition variables allows each entry of the SEM matrices to be an arbitrary function of multiple parameters and the person-specific definition variables of the value x_i. The only constraint is that for every parameter value θ, and every person, as represented by their corresponding definition variables value x_i, the result has to be a Gaussian. I will call the resulting method *extended SEM* (Neale et al., 2016). It is covered by the formalism introduced for definition variables.

3.2.2. Frequentist Inference

Inference in the context of SEM refers to making probabilistic statements about the parameters of a SEM. The frequentist approach to inference still dominates SEM. I will briefly explain how this inference procedure is typically applied to SEMs. For simplicity, I will restrict the presentation to standard SEMs. All introduced procedures can be expanded to be applicable to extended SEMs.

Point Estimation

The most commonly used point estimator for SEMs is the ML estimator. The ML estimate $\hat{\theta}$ for any SEM, as represented by parameterized mean vector $\mu(\theta)$, covariance matrix $\Sigma(\theta)$, and corresponding parameter space Θ, and given a data set $\mathbf{y} = \{y_i : i \in 1, \ldots, N\}$, is

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \prod_{i=1}^{N} \mathcal{N}(y_i; \mu(\theta), \Sigma(\theta)).$$

Hypothesis Testing

By far the most used hypothesis test for SEM is the likelihood-ratio test. The test statistic for the likelihood-ratio test is the likelihood-ratio statistic.

Definition 3.2.2. Let $\mathcal{M} = \{p(\mathbf{y}; \theta) : \theta \in \Theta\}$ be a parametrical statistical model, $H_0 : \theta^* \in \Theta_R \subset \Theta$ a null hypothesis, and $\mathbf{y} = \{y_i : i \in 1, \ldots, N\}$ a data set. Furthermore, let $\hat{\theta}$ be the unrestricted ML estimate, that is, the parameter value $\hat{\theta} \in \Theta$ that maximizes the likelihood $p(\mathbf{y}; \theta)$, and $\hat{\theta}_R$ the restricted ML estimate, that is, the parameter value $\hat{\theta} \in \Theta_R$ that maximizes the likelihood $p(\mathbf{y}; \theta)$, then

$$T(\mathbf{y}) = -2 \log \left(\frac{p(\mathbf{y}; \hat{\theta}_R)}{p(\mathbf{y}; \hat{\theta})} \right)$$

is called likelihood-ratio statistic.

The restricted parameter space Θ_R is a real subset of the parameter space Θ. Therefore, the fraction $p(\mathbf{y}; \hat{\theta}_R)/p(\mathbf{y}; \hat{\theta})$ is always smaller than 1, and, consequently, the likelihood-ratio statistic is always greater than 0. The larger the likelihood-ratio statistic, the greater the discrepancy between the likelihood given the null hypothesis and the likelihood given the full model, and, thus, the more evidence there is for rejecting the null hypothesis.

To obtain a proper statistical test based on the likelihood-ratio statistic a critical value c_α is needed such that the statistical test with the rejection region $R = \{T(\mathbf{y}) > c_\alpha\}$ has size α. The strategy to obtain the critical value c_α is to derive the distribution as represented by the pdf $p(T(\mathbf{y}))$ of the test statistic $T(Y)$ if the null hypothesis is true. By then choosing c_α such that

$$\int_{c_\alpha}^{\infty} T(\mathbf{y}) p(T(\mathbf{y})) \mathrm{d}T(\mathbf{y}) = \alpha$$

or equivalently

$$\int_{-\infty}^{c_\alpha} T(\mathbf{y}) p(T(\mathbf{y})) \mathrm{d}T(\mathbf{y}) = 1 - \alpha \tag{3.2.6}$$

one obtains a test of size α. For a limited class of statistical models and corresponding null hypotheses, the distribution of the test statistic $T(Y)$ under the null hypothesis can be derived. However, the exact distribution is not known for the general case.

But, the distribution of the likelihood-ratio test statistic $T(Y)$ under the null hypothesis can be approximated using only a fairly general set of conditions (Taboga, 2012c). One of the two major conditions is that both the restricted ML estimator $\hat{\theta}_R(Y)$ and the unrestricted ML estimator $\hat{\theta}(Y)$ are asymptotically distributed according to a Gaussian. This is typically fulfilled for a large enough data set $\mathbf{y}$ if the observations y_i are iid, as is the case for SEM. Multiple other rather technical conditions have to fulfilled for the maximum likelihood estimators to be approximately Gaussian (Taboga, 2012d). For SEMs, these conditions are typically met.

The second major condition is that the null hypothesis is nested within the statistical model, which is defined as follows.

Definition 3.2.3. Let $\mathcal{M} = \{p(y;\theta) : \theta \in \Theta\}$ be a parametrical statistical model with corresponding P-dimensional parameter space $\Theta \subset \mathbb{R}^P$, and null hypothesis $H_0 : \theta^* \in \Theta_R$. The null hypothesis H_0 is nested within the statistical model $\mathcal{M}$ iff a function $g : \mathbb{R}^P \to \mathbb{R}^Q$ with $Q \leq P$ exists such that the restricted parameter space Θ_R can be represented as follows:

$$\Theta_R = \{\theta : \theta \in \Theta \wedge g(\theta) = 0\}$$

The nested hypothesis H_0 is said to have Q degrees of freedom difference as compared to the statistical model $\mathcal{M}$.

For nested hypotheses the following theorem derives the approximate distribution of the test statistic $T(Y)$ under the null hypothesis.

Theorem 3.2.4. Let $T(Y^{(N)})$ be the random variable that describes the distribution of the likelihood-ratio test statistic for data sets of size N. If the null hypothesis $g(\theta) = 0$ is true, g is differentiable with a Jacobian matrix $J_g(\theta)$ of full rank Q, and the requirements for both the unrestricted und the restricted ML estimator to have asymptotic Gaussian distribution (see Taboga, 2012c) are fulfilled, then the distribution of likelihood-ratio test statistic

$$T(Y^{(N)})$$

converges in distribution (Taboga, 2012a) to a Chi-squared distribution with Q degrees of freedom. In shorthand one writes $T(Y^{(N)}) \xrightarrow{d} \chi^2(Q)$.

Knowing the approximate distribution of $T(Y^{(N)})$ under the null hypothesis, a hypothesis test with approximate size α can be constructed (see also Equation 3.2.6).

Theorem 3.2.5. Let $\mathcal{M}$ be a parametrical statistical model, $H_0 : \theta^* \in \Theta_R$ a nested hypothesis with Q degrees of freedom difference, $\mathbf{y} = \{y_i : i \in 1, \ldots, N\}$ a data set, and $T(\mathbf{y})$ the corresponding likelihood-ratio statistic, then the critical value c_α such that the statistical test with rejection region $\{T(\mathbf{y}) > c_\alpha\}$ has the size α is

$$c_\alpha = F_Q^{-1}(1 - \alpha),$$

where F_Q^{-1} is the inverse of the cumulative distribution function of a Chi-squared distribution with Q degrees of freedom.

The statistical test using the likelihood-ratio statistic in correspondence with the critical value from the last theorem is known as the likelihood-ratio test or Chi-squared test.

Confidence Regions

There are two common approaches to obtaining confidence regions for SEMs, Wald-type and likelihood-based. While Wald-type confidence regions were more popular historically, it has recently been argued that likelihood-based confidence regions should be favored (Pek & Wu, 2015). Thus, I will only present likelihood-based confidence regions.

The main idea of likelihood-based confidence regions is to use the likelihood-ratio test to construct confidence regions for the values of a parameter vector θ and confidence intervals for the values of a parameter θ_p. As I have shown in Theorem 2.3.15, confidence regions can be constructed from hypothesis tests for point hypotheses $\theta^* = \theta$. The idea is to build a confidence interval such as if we would test every point $\theta \in \Theta$ of a parameter space using a hypothesis test with corresponding null hypothesis $H_0 : \theta^* = \theta$ of size α. If the point is rejected, it is not included in the confidence region. This procedure produces a confidence region with confidence level $1 - \alpha$.

A point hypothesis $\theta^* = \theta$ is a special case of a nested hypothesis. Thus, the likelihood-ratio test can be used to generate confidence regions.

In Remark 2.3.16 I have elaborated how Theorem 2.3.15 can be extended to obtain confidence intervals for the values of a single parameter θ_p in a multi-parameter model. Essentially, the same strategy can be used as for confidence regions, but instead of a statistical test for the point hypothesis $\theta^* = \theta$, a statistical test for the hypothesis $\theta_p^* = \theta_p$ is required. Hypotheses of the form $\theta_p^* = \theta_p$ that only fix one parameter are also nested hypotheses. Thus, confidence intervals can also be constructed using the likelihood-ratio test.

3.2.3. Model Validation and Selection

As in the case of all statistical inference methods (see Section 2.7.1), model validation in the context of SEM refers to validation of the assumption that the statistical model is correctly specified. The approach originally proposed to assert this assumption for SEMs is the hypothesis test approach (see also Section 2.7.1). The model is interpreted as null hypothesis within the larger model of "all Gaussian distributions." If the null hypothesis is not rejected, it is assumed that the model is correctly specified. Thus, this approach assumes that the true distribution is a Gaussian distribution without assessing this assumption. The likelihood-ratio test statistic for this hypothesis test is

$$\chi^2 = -2 \log \left(\frac{\sup_{(\mu,\Sigma)\in\Theta_R} \mathcal{N}(\mathbf{y}; \mu, \Sigma)}{\sup_{(\mu,\Sigma)\in\Theta} \mathcal{N}(\mathbf{y}; \mu, \Sigma)} \right). \tag{3.2.7}$$

Θ_R contains all combinations of the mean vector μ and covariance matrix Σ that the proposed SEM allows, and Θ contains all μ and Σ combinations of the appropriate size.

Besides it being generally problematic to use hypothesis tests for model validation (see Section 2.7.1), a further problem of using the likelihood-ratio test for model evaluation is that the null hypothesis is almost always rejected, since the χ^2 value is highly sensitive to the sample size N (Little, 2013, Chapter 4, Section "Statistical Rationale"). This, however, is not a defect of the likelihood-ratio test. It is simply a consequence of the fact that statistical power increases with sample size and that virtually all SEMs are misspecified.

As a remedy, multiple other approaches, so-called fit indices, have been developed to validate SEMs. Almost all fit indices involve a trade-off between goodness of fit (i.e., the likelihood of the data at the ML estimate) and model complexity. For an introduction see, e.g., Little (2013, Chapter 4) or Kaplan (2009, Chapter 6). However, the problem with fit indices is that none of them accesses the correct specification of a model (Barrett, 2007). They are simply heuristics for scoring statistical models. Despite this fact, in practice, fit indices are commonly used to validate SEMs. Given thresholds of acceptable fit that are results of negotiations in a particular field, the model is then used as if it were certain that it is correctly specified.

Model selection is closely related to model validation. Instead of validating one particular model, a best of a set of models is selected. As such, the fit indices that are used to validate a SEM can also be employed to select between a set of competing SEMs. One simply selects the SEM with the highest score.

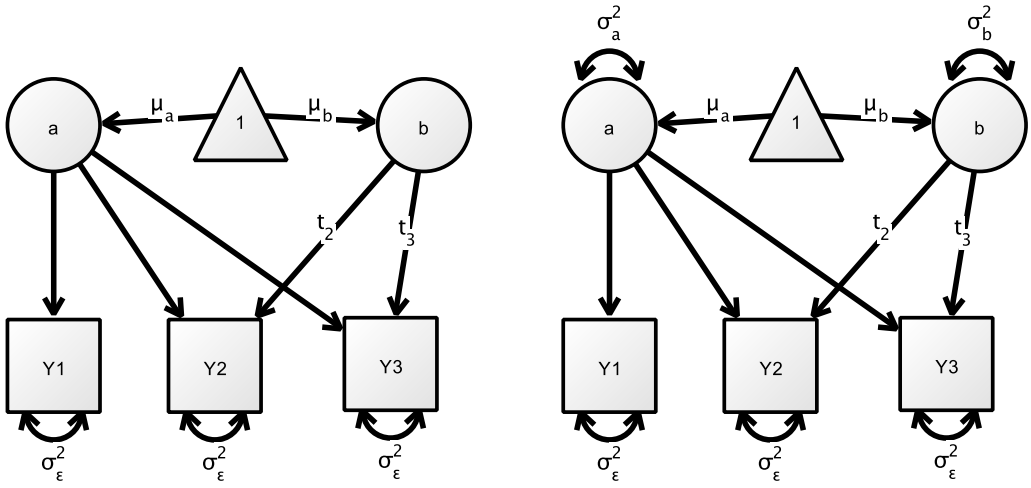
One alternative approach for model selection in the context of SEM is the hypothesis testing approach (see Section 2.7.1). That is, one starts with a basic model and extends the basis model as long as the model fit is improved. A likelihood-ratio test judges whether the model fit is improved by the extension. The extended model takes the role of the statistical model and the restricted model represents the null hypothesis. As a consequence, the restricted model has to be nested in the extended model.

3.2.4. Longitudinal Structural Equation Modeling

In longitudinal SEM, SEM is used for the analysis of panel data. For extended coverage of the topic, see Little (2013). I will limit the presentation to one example, namely the LGCM, which is particularly relevant for developmental and lifespan psychology, as it allows modeling of development using individual trajectories, thus, capturing between-person differences in development.

Assume that a univariate panel data set $\mathbf{y} = \{y_i : 1, \ldots, N\}$ has been observed. Every observation y_i is a realization of a random vector Y_i and contains all data for person i. The jth entry Y_{ij} of of the random vector Y_i corresponds to the jth observation for person i.

SEM can be used to specify a model for one person as represented by the random vector Y_i. The model for all persons follows from the iid assumption. One straightforward model



(a) Fixed effects LGCM (b) Regular LGCM

Figure 3.1.: Path diagrams displaying two LGCMs for three time points with uncorrelated latent factors and homogeneous residual error variances. Panel (a) displays the fixed effects LGCM that does not allow any variation in the slope and intercept parameters between persons. Panel (b) shows the LGCM that allows for variation in both parameters. a refers to the latent variable representing the intercept and b to the latent variable representing the slope. μ_a refers to the mean intercept and μ_b to the mean slope. In the case of the fixed effects LGCM, the means correspond to the value of each person, since there is no variation between persons. σ_a^2 and σ_b^2 refer to the variance of the mean and the slope respectively. σ_ϵ^2 denotes the measurement error variance. t_2, t_3 encode the time points of the respective observation. The time point of the first observation is implicitly encoded. Finally, Y_1, Y_2, Y_3 are the observed variables, each corresponding to one time point.

for a person's data y_i is to assume a linear trend. That is,

$$Y_{i,j} = a + bt_j + \epsilon_{i,j}, \tag{3.2.8}$$

with $a \in \mathbb{R}$ representing the intercept, $b \in \mathbb{R}$ representing the slope, $t_j \in \mathbb{R}$ encoding the time point of the jth measurement, and $\epsilon_{i,j} \sim \mathcal{N}(0, \sigma_\epsilon^2)$ representing the measurement error. Figure 3.1a displays the corresponding path diagram of the resulting SEM for three time points.

It is common to allow the intercept a as well as the slope b to vary between persons. This is achieved by allowing each person to have their individual slope and intercept parameter:

$$Y_{i,j} = a_i + b_i t_j + \epsilon_{i,j}. \tag{3.2.9}$$

Additionally, it is assumed that the person-specific parameters a_i, b_i are realizations of a Gaussian between-person distribution. Formally, this can be equivalently but more compactly represented by stating that every person-specific parameter a_i, b_i is distributed as follows:

$$a_i \sim \mathcal{N}(\mu_a, \sigma_a^2), b_i \sim \mathcal{N}(\mu_b, \Sigma_b^2). \tag{3.2.10}$$

Correlation between a_i and b_i can also be modeled.

This is also a SEM, commonly known as the LGCM. The corresponding path diagram is shown in Figure 3.1b. In the standard SEM representation of the LGCM the jth observation has to be made at the same time point for every person. Definition variables can be used to avoid this restriction. For this, the quantities t_j that encode the time point of the measurements are included as definition variables $t_{i,j}$.

3.3. Gaussian Process Regression

3.3.1. Weight-Space View

In this section, I will present GPR, the Bayesian regression technique on which GPPM is based. In the last section, I have shown that SEM can be seen as an extension of the GLM insofar that it allows the formulation of a set of linear equations instead of only one. The so-called weight-space view introduced in this section allows interpretation of GPR as a further extension of the GLM. In contrast to SEM, GPR only allows one linear equation.

GPR is motivated by the fact that the GLM can only represent linear relationships between the IVs and the DV. In the real world, however, many nonlinear relationships exist. The so-called basis function approach is to extend the GLM allowing nonlinear relationships. A basis function ϕ maps the independent variables into a different space. That is, instead of finding a function of the form $f(x) = x^\top \beta$, one obtains a function of the form

$$f(x) = \phi(x)^\top \beta,$$

where the basis function ϕ may describe any function. Thus, the basis function approach largely increases the set of statistical models that can be represented. However, model specification necessarily becomes more complex, since in addition to selecting the variables one of infinitely many basis functions has to be chosen. The interpretation of the parameter estimates becomes more difficult due to the introduction of nonlinearities.

GPR is commonly used for supervised learning problems. Typically, the data sets are large as compared to psychological data sets. Given large data sets, a model that allows for nonlinearities usually results in a more accurate prediction function. Also, since interpretation of the parameter estimates is typically not of interest, the increased difficulty of interpretation is no problem.

The derivation of the ML estimate and the remaining statistical inference procedures remain unchanged. Every observed value x of the IVs simply needs to be replaced by its image $\phi(x)$. As a result, a data set $D = \{(x_i, y_i) : i \in 1, \ldots, N\}$ is transformed to

$\{(\phi(x_i), y_i) : i \in 1, \ldots, N\}$. As shorthand for all observations of IVs in a data set, I again use

$$X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{bmatrix}.$$

For the image of all IVs X in a data set, I use the abbreviation

$$\Phi(X) = \begin{bmatrix} \phi(x_1)^\top \\ \vdots \\ \phi(x_N)^\top \end{bmatrix},$$

which I will further abbreviate as Φ in the following. The modified statistical model for the random vector Y of which $\mathbf{y} = \{y_i : i \in 1, \ldots, N\}$ is a realization is:

$$Y|\Phi \sim \mathcal{M} = \{N(\Phi\beta, \sigma_\epsilon^2 I_N) : (\beta, \sigma_\epsilon^2) \in \mathbb{R}^P \times \mathbb{R}_0^+\}.$$

P refers to the dimensionality of the weight vector β and thus, to the dimensionality of the transformed IVs $\phi(x)$.

In GPR, Bayesian inference is commonly used. In order to perform Bayesian inference on the statistical model $\mathcal{M}$, a prior on the parameters β and σ_ϵ^2 is needed. In GPR, it is commonly assumed that the prior for the weight vector is a Gaussian, i.e., $p(\beta) = \mathcal{N}(\beta; \mu_p, \Sigma_p)$. The error variance σ_ϵ^2 is assumed to be part of the statistical model. Thus, the statistical model changes to

$$\mathcal{M} = \{N(\Phi\beta, \sigma_\epsilon^2 I_N) : \beta \in \mathbb{R}^P\}.$$

Remember, in Bayesian inference, the pdf describing the statistical model is expressed as a conditional pdf: $p(\mathbf{y}|\Phi, \beta) = \mathcal{N}(\mathbf{y}; \Phi\beta, \sigma_\epsilon^2 I_N)$. By using Bayes' rule, one obtains the posterior distribution of the parameter

$$p(\beta|\Phi, \mathbf{y}) = \frac{p(\mathbf{y}|\Phi, \beta)p(\beta)}{p(\mathbf{y}|\Phi)},$$

with $p(\mathbf{y}|\Phi) = \int_{\mathbb{R}^P} p(\mathbf{y}|\Phi, \beta)p(\beta)\mathrm{d}\beta$, which follows from $p(\mathbf{y}, \beta|\Phi) = p(\mathbf{y}|\Phi, \beta)p(\beta)$ and marginalization.

Since GPR is mainly used in supervised machine learning, the posterior over the weight vector β is just an intermediate step. The main goal is to obtain a distribution for function values $f(x^*)$ at a set of values of the IVs that have not been observed for making predictions. I summarize a set of new IVs variables as matrix

$$\Phi^* = \begin{bmatrix} \phi\left(x_1^*\right) \\ \vdots \\ \phi\left(x_{N_2}^*\right) \end{bmatrix}.$$

In the language of machine learning this set is the holdout set whereas the data set $(X, \mathbf{y})$ used to obtain the posterior distribution of the parameters is the training set. The posterior distribution for the holdout set predictions

$$f(\Phi^*) = \begin{bmatrix} f(\phi(x_1^*)) \\ \vdots \\ f(\phi(x_{N_2}^*)) \end{bmatrix}$$

is obtained by linking the posterior distribution over the weights $p(\beta|\Phi, \mathbf{y})$ with the likelihood function $p(f(\Phi^*)|\Phi^*, \beta) = \mathcal{N}(f(\Phi^*); \Phi^*\beta, \sigma_\epsilon^2 I_N)$ implied by the statistical model. It is

$$p(f(\Phi^*)|\Phi, y, \Phi^*) = \int_{\mathbb{R}^P} p(f(\Phi^*)|\Phi^*, \beta)p(\beta|\Phi, \mathbf{y})\mathrm{d}\beta,$$

and commonly known as predictive distribution. I will derive an analytical expression for the predictive distribution in the next section.

3.3.2. Function-Space View

The function-space of GPR is essentially equivalent to the weight-space view. However, it introduces a central ingredient of this thesis, namely the Gaussian process (GP). It also allows description of the statistical model and the prior in an alternative, more compact fashion.

The prior over weights $p(\beta) = \mathcal{N}(\beta; \mu_p, \Sigma_p)$ can be transformed into a prior over functions. For any value of the IVs x, the random vector $f(\phi(x)^\top\beta)$ is distributed according to the following Gaussian distribution:

$$\mathcal{N}(\phi(x)^\top\mu_p, \phi(x)^\top\Sigma_p\phi(x)).$$

Thus, for any data set $D = (X, \mathbf{y})$, the prior over the corresponding functional values $f(X) = \Phi(X)\beta$ is

$$f(X) \sim \mathcal{N}(\Phi\mu_p, \Phi\Sigma_p\Phi^\top).$$

Therefore, the prior over functions implies a prior for every possible data set X. The prior over functions can be described by the implied expected prediction for the DV based on any value of the IVs x

$$\mathbb{E}[f(x)] = \phi(x)^\top\mu_p \tag{3.3.1}$$

and the implied covariance for every pair of values x, x'

$$\mathrm{Cov}(f(x), f(x')) = \phi(x)^\top\Sigma_p\phi(x')^\top. \tag{3.3.2}$$

Hence, the indirect prior over weights can be transformed into a direct prior over functions. However, in contrast to the prior over weights, the prior over functions cannot be described by a multivariate Gaussian. The reason for this is that the set of possible values $\mathcal{X}$ that the IVs can obtain might be, and in most applications of GPR is, infinite.

The prior over function values is over the "random vector" $\{f(x) : x \in \mathcal{X}\}$, which as a consequence has infinite dimensionality. But, a random vector can not have infinite dimensionality. Hence, $\{f(x) : x \in \mathcal{X}\}$ is not a random vector. It is a so-called *stochastic process*.

Definition 3.3.1. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, then a *stochastic process* is a collection of random variables $\{f(x) : x \in \mathcal{X}\}$ indexed by an index set $\mathcal{X}$ where each $f(x)$ is a random variable on the sample space Ω.

The prior over functional values $\{f(x) : x \in \mathcal{X}\}$ is indeed a particular kind of a stochastic process, namely a *Gaussian process (GP)*.

Definition 3.3.2. A *Gaussian process (GP)* is a stochastic process for which any finite subset of $\{f(x) : x \in \mathcal{X}\}$ (which is a random vector) is distributed according to a Gaussian distribution.

Remark 3.3.3. Because the index set for all GPs used in this work and the set of possible values for the vector of IVs are identical I will use the same symbol for them even though, in general, they are distinct concepts.

Instead of being described by a mean vector and a covariance matrix, a GP is described by a mean function $m(x)$ and a covariance function $k(x, x')$ (called kernel function or simply kernel in machine learning). Equations 3.3.1 and 3.3.2 are examples for a mean and a covariance function. It is convenient to extend the mean and covariance functions to subsets of the index set $\mathcal{X}$.

Definition 3.3.4. For any two subsets of the index set $X, X^* \subseteq \mathcal{X}$ of size $|X| = N_1, |X| = N_2$, mean function $m(x)$ and covariance function $k(x, x')$, let

$$M(X) = [m(x_1), m(x_2), \ldots, m(x_{N_1})]^\top$$

be the mean of the random vector $\{f(x) : x \in X\}$ implied by the mean function $m(x)$, and

$$K(X, X^*) = \begin{bmatrix} k(x_1, x_1^*) & k(x_1, x_2^*) & \ldots & k(x_1 x_{N_2}^*) \\ k(x_2, x_1^*) & k(x_2, x_2^*) & & \vdots \\ \vdots & & \ddots & \\ k(x_{N_1}, x_1^*) & \ldots & & k(x_{N_1}, x_{N_2}^*). \end{bmatrix}$$

be the cross-covariance matrix of the random vectors $\{f(x) : x \in X\}$ and $\{f(x^*) : x^* \in X^*\}$.

Remark 3.3.5. Note that $K(X, X)$ describes the covariance matrix of the random

vector $\{f(x) : x \in X\}$.

Using this notation, one can state which properties two functions $m(x)$ and $k(x, x')$ need to fulfill to describe a valid GP, i.e., to be valid mean and covariance functions.

Theorem 3.3.6. The triple of an index set $\mathcal{X}$, mean function $m : \mathcal{X} \to \mathbb{R}$ and covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ describes a GP iff $M(X)$ is a valid mean vector and $K(X, X)$ is a valid covariance matrix for every subset $X \subset \mathcal{X}$.

The transformation of the prior over weights into a prior over functional values makes it possible to obtain the predictive distribution $p(f(X^*)|y, X, X^*)$ in one step. Instead of the two-step approach, first finding the posterior distribution $p(\beta|X, \mathbf{y})$ based on the data set $D = (X, \mathbf{y})$, and then obtaining the predictive distribution by linking it with the likelihood.

Expressing the prior as a GP over function values allows formulation of the prior distribution of training predictions $f(X) = \Phi(X)\beta$ and test predictions $f(X^*) = \Phi(X^*)\beta$. Let m and k be the mean and the covariance functions that describe the GP prior, then the joint distribution of $f(X)$ and $f(X^*)$ is

$$\begin{bmatrix} f(X) \\ f(X^*) \end{bmatrix} \Bigg| X, X^* \sim \mathcal{N}\left(M\left(\begin{bmatrix} X \\ X^* \end{bmatrix} \right) \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \right).$$

By describing the prior over functions, the likelihood, which maps the predictions $f(X)$ to the observations $\mathbf{y}$, changes to $p(\mathbf{y}|f(X)) = \mathcal{N}(f(X), \sigma_\epsilon^2 I_N)$. Thus, the random vector Y of which $\mathbf{y}$ is one realization can be described as $Y = f(X) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 I_N)$. Note that the prior over functions and the error variance describes the statistical model completely. It follows that

$$\begin{bmatrix} Y \\ f(X^*) \end{bmatrix} \Bigg| X, X^* \sim \mathcal{N}\left(M\left(\begin{bmatrix} X \\ X^* \end{bmatrix} \right) \begin{bmatrix} K(X, X) + I_n \sigma_n^2 & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \right). \tag{3.3.3}$$

Finally, the predictive distribution is obtained by conditioning the joint distribution in Equation 3.3.3 on the training targets $\mathbf{y}$. It is:

$$\begin{aligned} f(X^*)|X, X^*, \mathbf{y} &\sim \mathcal{N}(\mathbb{E}(f(X^*)|X, X^*, \mathbf{y}), \mathrm{Cov}(f(X^*)|X, X^*, \mathbf{y})), \text{ with} \\ \mathbb{E}(f(X^*)|X, X^*, \mathbf{y}) &= M(X^*) + K(X^*, X)[K(X, X) + \sigma_\epsilon^2 I_N]^{-1}(\mathbf{y} - M(X)) \\ \mathrm{Cov}(f(X^*)|X, X^*, \mathbf{y}) &= K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma_\epsilon^2 I_N]^{-1} K(X, X^*). \end{aligned}$$

3.3.3. Model Selection

Model selection in the context of GPR refers to selecting a GPR model, that is, a mean, a covariance function, and the amount of measurement error σ_ϵ^2. Up to this point it was assumed that these three quantities were given. These quantities encode the statistical model as well as the prior of its parameters. Thus, they encode a Bayesian model. In

GPR one popular approach to select a model is model evidence maximization (Bishop, 2006, Section 3.4).

The *model evidence* or *marginal likelihood* is the probability of the data given the model. In the following I will provide an explanation as to why a model section based on the marginal likelihood is reasonable and how the marginal likelihood is computed. For alternative treatments, see Bishop (2006, Section 3.4) in the context of MLR, Claeskens and Hjort (2008, Section 3.2) with an emphasis on the connection to the Bayesian information criterion (BIC) or Rasmussen (2006, Section 5.2) in the context of GPR. An alternative reason for using the model evidence for model selection to the one presented here is that it corresponds to choosing the model with the highest probability given the data under a uniform model prior. Thus, it corresponds to the Bayesian standard procedure of selecting the model with the highest probability given the data, as introduced in Section 2.7.2.

Let H be a random vector that encodes the chosen model and h a realization that describes one model. h encodes a particular statistical model $\{p(y|\theta) : \theta \in \Theta_h\}$, with corresponding model-specific parameter space Θ_h, as well as the corresponding prior $p(\theta|h)$. Equivalently, a model can encode a mean and a covariance function. Let the index set Θ describe all possible probability distributions instead of a parameter space as usual. If one sets $p(\theta|h) = 0$ for all parameter values $\theta \in \Theta$ that are not within Θ_h, the statistical model can be completely described as a prior distribution $p(\theta|h)$ over all possible distributions. Using this formalism, Bayes' rule (displayed in Equation 2.4.2) can be re-expressed as the following to emphasize that all inferences made are conditional on the choice of the model:

$$p(\theta|y, h) = \frac{p(y|\theta, h)p(\theta|h)}{p(y|h)}.$$

In a so-called fully Bayesian treatment one would impose an unconditional prior $p(h)$ over models to be able to marginalize over all possible models. Every model h comes with its own parameter space Θ_h and corresponding prior $p(\theta|h)$. Remember that every parameter value corresponds to a probability distribution. Thus, specifying a prior over models and a prior over parameters for each model is simply a hierarchical way of specifying a prior over all probability distributions. Mathematically, this can be expressed as

$$p(\theta) = \int p(\theta|h)p(h)\mathrm{d}h.$$

Using this prior, Bayes' rule can be written hierarchically:

$$p(\theta|y) = \frac{p(y|\theta)\int p(\theta|h)p(h)\mathrm{d}h}{\int p(y|h)p(h)\mathrm{d}h}.$$

The fully Bayesian approach is computationally unfeasible, since it requires integration over all possible probability distributions. Additionally, a prior over all possible probability distributions is needed, which is also impractical for most problems.

Therefore, an empirical Bayesian treatment is often used in practice. It approximates the fully Bayesian posterior $p(\theta|y)$ by

$$\frac{p(y|\theta,h^*)p(\theta|h^*)}{p(y|h^*)},$$

where h^* is the model that maximizes the model evidence

$$p(y|h) = \int p(y|\theta)p(\theta|h)\mathrm{d}\theta.$$

The rationale for using this strategy is that for a flat model prior $p(h)$, the posterior distribution based on the model with the highest evidence is a good approximation if the model evidence $p(y|h)$ is highly peaked at its maximum $p(y|h^*)$.

Given a data set $D = (X, \mathbf{y})$, the model evidence for a given GP prior

$$f(x) \sim \mathcal{GP}(m(x), k(x,x))$$

and measurement error σ_ϵ^2 is

$$p(\mathbf{y}|X, m, k, \sigma_\epsilon^2) = \mathcal{N}(\mathbf{y}; M(X), K(X,X) + I_N\sigma_\epsilon^2).$$

Typically for GPR, the set containing the candidate models is not finite, but uncountably infinite. The model set can commonly be described by a parameterization of the mean $m(x;\theta)$ and the covariance function $k(x,x';\theta)$. Note that every value of the parameter θ now refers to a statistical model instead of a parameter value within a statistical model (i.e., a distribution). Therefore, the parameters of the mean and covariance functions are called hyper-parameters.

A simple example for a parameterized mean function is the so-called constant mean function

$$m(x;c) = c,$$

with $c \in \mathbb{R}$. Probably the most prominent parameterized covariance function is the exponential squared covariance function

$$k(x,x';[\sigma,l]) = \sigma^2 \exp\left(\frac{(x-x')^2}{-2l^2}\right),$$

with $\sigma, l \in \mathbb{R}^+$, which is often inappropriately called exponential squared covariance.

For a given data set $D = \{(x_i, y_i) : i \in 1,\ldots,N\}$, the uncorrelated measurement error can be included into the parameterization of the covariance function by using $k_y(x,x';[\theta,\sigma_\epsilon^2]) = k(x,x';\theta) + \delta(x-x')\sigma_\epsilon^2$, where $\delta(x)$ is the Dirac delta function, i.e., it is 0 everywhere but at 0. If the set of candidate models used for model selection via model evidence maximization can be described by the parameterized mean $m(x;\theta)$ and covariance function $k_y(x,x';\theta)$ with a corresponding parameter space Θ, the selected model is represented by a parameter value $\hat{\theta}$, that is,

$$\hat{\theta} = \underset{\theta\in\Theta}{\arg\max}\, \mathcal{N}(\mathbf{y}, M(X;\theta), K_y(X,X;\theta)).$$

3.3.4. Model Validation

GPR is typically used as a supervised learning method. In this context, model validation refers to estimating the supervised learning risk of the prediction function $f(x) = y$ (see Section 2.7.1).

GPR returns a predictive distribution $p(y|x, D)$ instead of a prediction function. One can extract a prediction function from the predictive distribution by minimizing the Bayesian expected loss (see Definition 2.4.1) for a given loss function. If the true target value y is known, the loss of making the prediction $f(x)$ is $L(y, f(x))$. However, the true value y is unknown. Thus, the expectation with respect to the posterior distribution $p(y|x, D)$ is used instead. That is, the action with the smallest Bayesian expected loss is chosen. The Bayesian expected loss for a prediction function $f(x)$ under the posterior $p(y|x, D)$ with loss function L is

$$R_B(f(x), p, x) = \int L(y, f(x))p(y|x, D)\mathrm{d}y.$$

Thus, the prediction function $\hat{f}(x)$ that minimizes this is

$$\hat{f}(x) = \arg\min_{f(x)} R_B(f(x), p, x).$$

If the squared loss is used, the optimal $\hat{f}(x)$ is

$$\hat{f}(x) = \mathbb{E}(Y|x, D) = \int_{\Omega_Y} yp(y|x, D)\mathrm{d}y.$$

The supervised learning risk of the optimal prediction function $\hat{f}(x)$ can now be estimated as for any other prediction function (see Section 2.5).

Alternatively, there are ways to validate the predictive distribution directly (Rasmussen, 2006, Section 5.4).

3.4. Gaussian Process Time Series Modeling

3.4.1. Foundations

In this chapter, I adapt GPR for using it for panel data. These can be interpreted as an extension of time series data. Instead of one time series originating from one person, they consist of multiple time series, each originating from a different person.

GPR has already been adapted for the analysis of time series data (see, e.g., Roberts et al., 2013). Thus, as the first step, I will review using GPR to analyze time series data in this section, which I will refer to as *Gaussian process time series modeling (GPTSM)*.

First, I will briefly recapitulate how GPR is typically used. Given a set of IVs for an object a DV of the object is predicted. An application example within psychology could be the use of genetic data to predict the general intelligence factor g of a person.

GPR can be used for modeling time series data as follows. Psychological time series typically consist of repeated measures of some properties of a single person. An example for a psychological time series is the general intelligence factor g of a person being measured repeatedly at different ages across the life-span. While not belonging to the standard time series modeling approaches, using GPR for the analysis of time series data has already received some attention. For example, it has been used to model data from a weather sensor network (Roberts et al., 2013), water levels of the Nile (Roberts et al., 2013), the development of stock indices (Damouras, 2008; Roberts et al., 2013), the brightness of stars (Roberts et al., 2013), respiration data (Brahim-Belhouari & Bermak, 2004), the trajectory of a computer mouse controlled by a participant in a cognitive science experiment (Cox et al., 2012), human arm movements (Cunningham et al., 2012), the number of spots on the sun (Damouras, 2008), clinical time series such as the blood parameters after surgery (Liu, Wu, & Hauskrecht, 2013), the trajectory of the dance of honey bees (Saatçi et al., 2010), and snowfall in Canada (Saatçi et al., 2010).

I will start by describing GPTSM for univariate time series. After that, I turn to GPTSM for multivariate time series data. In this case, univariate time series refers to the fact that their is only one DV. At least one IV exists, namely time. Further IVs can also be included. As long as there is only one DV, I still consider this a univariate time series, even though the actual data are multivariate, since they include at least one IV. Multivariate time series contain more than one DV.

In univariate time series analysis a time series $[y_1, \ldots, y_T] = \mathbf{y}$ with corresponding IVs $X = [x_1, \ldots, x_T]$ for each observation is observed. In the simplest case the IVs correspond to some representation of time (e.g., age). Besides time, the IVs may contain other information that explain the DV. The goal of time series analysis is to make predictions for unobserved values y^* of the time series using the corresponding IVs value x^*, and to find the true generating distribution. This is different to conventional GPR where the true generating distribution is typically not of interest.

As for GPR, the starting point of GPTSM is a set of models as represented by a parameterized GP

$$\{\mathcal{GP}(m(x;\theta), k(x;x';\theta)) : \theta \in \Theta\},$$

with a corresponding index set $x, x' \in \mathcal{X}$. To choose the hyper-parameter value θ one can proceed as in the conventional application of GPR and use model evidence maximization. Once the hyper-parameter has been identified, the predictive distribution for new time points x^* can be obtained as for GPR.

The problem with the inference approach used for GPR is that it does not directly infer the true generating distribution. Thus, in GPTSM another approach with a different interpretations of a GP is popular. In GPR, a GP is interpreted as a prior over functions, that is, as a compact representation of a statistical model and its corresponding prior. In this formalism, every GP corresponds to a particular statistical model and its corresponding prior. Selecting a GP out of a set of GPs using model evidence maximization is thus interpreted as model selection. However, for time series analysis a GP is often interpreted as a stochastic process of which the time series $y_1, \ldots, y_T$ is a realization.

Thus, a set of GPs can also be interpreted as a single statistical model instead of a set of statistical models. To emphasize this, I write

$$Y(x) \sim \mathcal{M} = \{\mathcal{GP}(m(x), k(x, x') : \theta \in \Theta\},$$

where $Y(x)$ refers to the random variables representing the hypothetical observation of the time series at the point described by the value x of the IVs. The ML estimate $\hat{\theta}_{\mathrm{ML}}$ of the parameters of the statistical model $\mathcal{M}$ is identical to the hyper-parameter value $\hat{\theta}$ that maximizes the model evidence of the set of models $\mathcal{M}$. In the following, I will use the interpretation that the set of GPs $\mathcal{M}$ is a statistical model. Taking this approach, the hyper-parameters should rather be called parameters again.

By imposing a prior on the parameters θ, standard Bayes' rule inference can be employed to compute the posterior:

$$p(\theta|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \theta)p(\theta)}{\int p(\mathbf{y}|X, \theta)p(\theta)\mathrm{d}\theta}.$$

The model evidence

$$p(\mathbf{y}|X, \theta) = \mathcal{N}(y; M(X; \theta), K(X, X; \theta))$$

as introduced for GPR acts as the likelihood function. The posterior distribution over stochastic processes expresses the belief which process might have generated the observed time series. The integral $\int p(\mathbf{y}|\theta, X)p(\theta)\mathrm{d}\theta$ is typically not exactly computable. However, various approximation techniques such as Markov Chain Monte Carlo techniques can be utilized (Roberts et al., 2013).

The predictive distribution for a new value of IVs x^* is obtained using the usual rule:

$$p(y^*|X, \mathbf{y}, x^*) = \int p(y^*|x^*, \theta)p(\theta|X, \mathbf{y})\mathrm{d}\theta,$$

where y^* refers to the hypothetical values of the random variable $Y(x^*)$. Note that the regular GPR inference approach is a special case of the approach discussed here. As prior $p(\theta)$, the delta prior (i.e., the prior that assigns all mass to one parameter value θ) maximizing the model evidence is used. In sum, using GPR for univariate time series analysis is relatively straightforward.

3.4.2. Extension to Multivariate Time Series

In a multivariate time series multiple variables are repeatedly observed. Using the approach developed in the previous section such time series can be modeled using GPTSM, as long as only one variable is treated as DV. However, many analyses performed in psychology require treatment of more than one variable as the DV. In a cross-lagged panel model (Burkholder & Harlow, 2003), for example, the lagged influence of at least two variables onto each other is of interest. To perform such analyses using GPTSM, this approach needs to be extended to allow for the simultaneous modeling of multiple DVs, each contributing one time series. I will call this approach multivariate GPTSM.

A straightforward approach to specifying a multivariate GPTSM is to give each DV a distinct label l, and to use the label as an additional IV. Using this approach, the covariance function is of the form $k([x, l], [x', l'])$. Thus, there is one joint covariance function for all time series, which can represent arbitrary relationships between and within the time series.

Separating the joint covariance function $k([x, l], [x', l'])$ into L autocovariance functions $k_l(t, t')$, one for each time series l, and $L(L-1)/2$ cross-covariance functions $k_{ll'}(x, x')$, one for each unique time series pair l, l', simplifies model specification. While this decomposition allows a relatively comprehensible description of the joint covariance function $k([x, l], [x', l'])$, it is hard to check whether a certain covariance function is valid. A necessary but not sufficient condition for the validity of the joint covariance function $k([x, l], [x', l'])$ is that all the autocovariance functions $k_l(x, x')$ are positive definite. Additionally to that, the cross-covariance functions $k_{ll'}(x, x')$ need to fulfill certain conditions that depend on the form of the autocovariance function (see Boyle, 2007, Section 3.1).

An unproblematic but less general way to define a valid joint covariance function $k([x, l], [x', l'])$ for a multivariate time series is the following approach, taken from Turner (2012, Section 3.6.2): A set of M independent latent GPs $\{g_i, i \in 1, \ldots, \mathrm{M}\}$, with typically $M \leq L$, is specified using arbitrary autocovariance functions $k_i(x, x')$. In this context, independence is identical to all cross-covariance functions $k_{ll'}(x, x')$ being 0. Then a matrix $A \in \mathbb{R}^{L \times M}$ is used to link the independent GPs to the potentially dependent unobserved true values of the multivariate time series $f(x) = Ag(x)$. The true values are again linked to the observed values by incorporating measurement error in the form

$$Y = f(x) + \epsilon_x \quad \epsilon_x \sim \mathcal{N}(0, \Sigma_\epsilon).$$

This always leads to a valid covariance function (see Turner, 2012, Section 3.6.2). However, taking this approach does not allow expression of every joint covariance function.

To summarize, while univariate GPTSM is relatively straightforward, multivariate GPTSM is more evolved, since the specification of the covariance function becomes more complex. Propositions that make specifying a multivariate GPTSM easier do exist, but they cannot express every multivariate GPTSM.

3.5. Gaussian Process Panel Models

3.5.1. Foundations

In the following sections, I will extend GPTSM for panel data. I call the resulting method *Gaussian process panel modeling* (GPPM).

I first introduce the necessary notation for panel data. In a panel data set, N time series $\{y_i \in 1, \ldots, N\}$ have been observed. Each time series y_i originates from one person. Each time series y_i contains observations $y_{ij} \in \mathbb{R}$ with corresponding IVs $x_{ij} \in R^K$. Multivariate time series are also covered within this formalism. Here, the IVs simply contain a label variable (see previous section).

One possibility to analyze panel data is to use GPTSM, and perform a separate analysis for each person, which means that for each person, the data of all others are ignored. This approach is reasonable if it can be assumed that people are not connected to each other. Formally, it is assumed that each person's time series is a realization of a GP such that

$$Y_i(x) \sim \mathcal{GP}(m_i^*(x), k_i^*(x, x')).$$

x refers to the value of the IVs describing the observation represented by the random vector $Y_i(x)$. The statistical inference task is to deduce the mean and covariance functions m_i^*, k_i^* for each person. This is carried out separately for each person.

If the assumption that there is no connection between people is not justified, some relationship between the person-specific GPs needs to be introduced. Arguably, the easiest approach is to assume that each person's time series is a realization of the same GP, and that the person-specific GPs are mutually independent (iid assumption). Note that this is not equivalent to assuming that there is no inter-individual variation. Indeed, I will show in Section 3.6 that many forms of inter-individual variation can be specified using this assumption. Also, this is the same approach as the one taken in SEM: One starts with a model for one person, and the model for everyone follows from the iid assumption.

If the generating GP is assumed to be the same for each person, it is justified to postulate the same statistical model

$$Y_i(x) \sim \{\mathcal{GP}(m(x), k(x, x')) : \theta \in \Theta\} \tag{3.5.1}$$

for every person, as expressed by a pair of parameterized mean and covariance functions. I call such a model a GPPM.

Equation 3.5.1 denotes a set of GPs. A GP can be interpreted as a infinite dimensional probability distribution. Classical statistical inference requires a statistical model that represents a set of finite dimensional distributions. For every finite, observed time series y_i a set of finite dimensional distributions is implied by a GPPM. Let T_i be the number of observed time points, and X_i a matrix with T_i rows where each row x_{ij} contains the IVs for the jth observation of person i, that is, for y_{ij}, then the statistical model for the time series y_i implied by the GPPM 3.5.1 is

$$p(y_i|X_i) \in \{\mathcal{N}(y_i; M(X_i;\theta), K(X_i, X_i;\theta)) : \theta \in \Theta\}.$$

The statistical model implied for a panel data set $D = (X, \mathbf{y})$, with $X = (X_1, \ldots, X_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$, follows from the mutual independence assumption and is

$$p(\mathbf{y}|X) \in \left\{ \prod_{i=1}^{N} \mathcal{N}(y_i; M(X_i;\theta), K(X_i, X_i;\theta)) : \theta \in \Theta \right\}.$$

This is a regular statistical model.

3.5.2. Model Specification

Model specification in the context of GPPM refers to the process of specifying a set of GPs that formalizes the assumptions about the process of interest. The first step is to define which IVs to use: The IV shared by all GPPMs is some representation of time. Besides that, arbitrary variables can be included as IVs.

The second important step to specify a GPPM is to define parameterized mean and covariance functions. There are many ways to derive the parameterized mean and covariance functions. One fruitful approach seems to be to (1) specify the person-level GPPM, describing the assumptions about the intra-individual variation, and to (2) use the rules that will be established in Section 3.6 to allow observed and unobserved inter-individual variation.

The specification of the person-level GPPM itself can also be challenging, because parameterized mean and covariance functions that express the assumptions about the process of interest need to be formulated. The following rule makes it possible to break down the specification of the mean and the covariance functions into smaller, better manageable components.

Theorem 3.5.1. Let W, Z be two independent GPPMs using the same IVs or more formally the same index set $x \in \mathcal{X}$

$$W_i(x) \sim \{\mathcal{GP}(m_w(x;\theta_w), k_w(x, x';\theta_w) : \theta_w \in \Theta_w\}$$
$$Z_i(x) \sim \{\mathcal{GP}(m_z(x;\theta_z), k_z(x, x';\theta_z) : \theta_z \in \Theta_z\},$$

then the weighted sum $Y = aW + bZ$ of the two models is a GPPM on the index set $\mathcal{X}$ of the form:

$$Y_i(x) \sim \{\mathcal{GP}(m_y(x;\theta_y), k_y(x, x';\theta_y)) : \theta_y \in \Theta_w \times \Theta_z\}$$
$$m_y(x;[\theta_w, \theta_z]) = a m_w(x;\theta_w) + b m_z(x;\theta_z)$$
$$k_y(x, x';[\theta_w, \theta_z])) = a^2 k_w(x, x';\theta_w) + b^2 k_z(x, x';\theta_z).$$

Proof: This follows from the fact that for two independent univariate Gaussian variables $W \sim (\mu_w, \sigma_w^2)$, $Z \sim (\mu_z, \sigma_z^2)$, the weighted sum $aW + bZ$ is distributed according

to a Gaussian with mean $a\mu_w + b\mu_z$ and variance $a^2\sigma_w^2 + b^2\sigma_z^2$. This in turn follows as a special case of the linear transformation theorem of a Gaussian random vector (Theorem A.3.8 in Appendix A.3). □

Example 3.5.2. An unrealistic assumption for illustrative purposes could be that each person's general intelligence factor g develops according to a linear trend across the lifespan. However, systematic within-day general intelligence g variation may also occur. The linear trend, including measurement error σ_ϵ^2, can be represented by the following GPPM. In this and the following GPPM, I refer to the value of the IV, as $s, t \in \mathbb{R}$ instead of as $x, x' \in \mathbb{R}^K$. This is commonly done if there is only one IV that is a representative of time.

$$W_i(t) \sim \{\mathcal{GP}(a + bt, \sigma_\epsilon^2) : (a, b, \sigma_\epsilon^2) \in \Theta_w\}$$

The systematic periodic changes can be represented by the GPPM

$$Z_i(t) \sim \left\{\mathcal{GP}\left(0, \sigma^2 \exp\left(\frac{-2\sin^2(\pi|s-t|/p)}{l^2}\right)\right) : (\sigma^2, p, l^2) \in \Theta_z\right\}.$$

The combination of linear change and systematic periodic fluctuations can be represented by the GPPM

$$Y_i(t) \sim \left\{\mathcal{GP}\left(a + bt, \sigma_\epsilon^2 + \sigma^2 \exp\left(\frac{-2\sin^2(\pi|s-t|/p)}{l^2}\right)\right) : (a, b, \sigma_\epsilon^2, \sigma^2, p, l^2) \in \Theta_y\right\}.$$

More combination rules for GPPMs can be derived from the combination rules for GPR models (e.g., Rasmussen, 2006, pp. 94–95). However, the sum rule seems to be the most useful for GPPMs, as indicated by the number of times it is employed in the remainder of this monograph.

3.6. Inter-Individual Variation in Gaussian Process Panel Models

Assuming that the data of every person constitute a realization of the same GP might seem like as if any inter-individual variation were disallowed. However, this is not the case. In this section, I will show this assumption allows incorporation of both observed and unobserved heterogeneity. In accordance with the literature, I use observed heterogeneity as a synonym for inter-individual variation that can be explained using IVs. Unobserved heterogeneity refers to inter-individual variation that is not explained by IVs, as typically modeled by the random effects approach.

3.6.1. Observed Heterogeneity

I formalize observed heterogeneity in GPPMs as follows. Let

$$Y_i(x) \sim \{\mathcal{GP}(m(x;\theta), k(x, x;\theta)) : \theta \in \Theta\}$$

be a GPPM. It is assumed that some additional IVs $\tilde{x}_i$ that are stable within a person have been observed. The IVs $\tilde{x}_i$ influence the assumption about the person-specific process and, thus, the value of some parameters within the parameter vector θ. This can be formalized by using a person-specific parameter θ_i that is computed by a deterministic function $f(\theta, \tilde{x}_i, \tilde{\theta})$, which in turn encodes the assumptions about the ways the IVs $\tilde{x}_i$ influence the parameters. The function itself also typically contains to-be estimated parameters $\tilde{\theta}$.

In the examples in this section the only IV that explains intra-individual variation is some representative of time. Thus, in accordance with the literature, I will refer to a pair of values for the IV as s, t, instead of as x, x'. Since the IV encodes time, $s, t \in \mathbb{R}$

Example 3.6.1. As a motivating example for modeling observed heterogeneity, consider the following GPPM, which encodes the assumption that all persons follow the same linear trend described by an intercept a and a slope b.

$$m(t; [a, b]) = a + bt \qquad k(s, t; \sigma_\epsilon^2) = \delta(s - t)\sigma_\epsilon^2$$

Imagine that this model describes the learning curve during a cognitive training study. At time point $t = 0$ the training begins. It may be unrealistic to assume that everybody's ability at the beginning a is the same. Additionally, the scalar-valued IV $\tilde{x}_i$, parents' income, has been measured. It is assumed that parents' income $\tilde{x}_i$ increases the starting ability a_i linearly, i.e.,

$$a_i = c + d\tilde{x}_i.$$

$c, d \in \mathbb{R}$ are scalar-valued parameters. Parents' income does not influence the steepness of the learning curve b or the amount of measurement error σ_ϵ^2. Thus, the function mapping the IV $\tilde{x}_i$ and the parameters $a, b, c, d, \sigma_\epsilon^2$ onto a person-specific parameter θ_i is

$$\theta_i = f([a, b, \sigma_\epsilon^2], \tilde{x}_i, [c, d]) = \begin{bmatrix} c + d\tilde{x}_i \\ b \\ \sigma_\epsilon^2 \end{bmatrix}$$

Note that only the starting level a is individualized. The function does not change the remaining parameters.

Observed heterogeneity, as defined here, can be incorporated into GPPMs. The parameter-influencing function $f(\theta, \tilde{x}_i, \tilde{\theta})$ simply implies a new mean $\tilde{m}$ and covariance function $\tilde{k}$. The new mean and covariance functions are as follows:

$$\tilde{m}([x_i, \tilde{x}_i]; [\theta, \tilde{\theta}]) = m(x_i; f(\theta, \tilde{x}_i, \tilde{\theta}))$$
$$\tilde{k}([x_i, \tilde{x}_i], [x_i', \tilde{x}_i]; [\theta, \tilde{\theta}]) = k(x_i, x_i'; f(\theta, \tilde{x}_i, \tilde{\theta})).$$

Thus, the parameters of the new GPPM become $[\theta, \tilde{\theta}]$ and the IVs $[x_i, \tilde{x}_i]$. It needs to be ensured that the function $f(\theta, \tilde{x}_i, \tilde{\theta})$ is such that the new mean and covariances function are still valid. However, this still allows for a vast number of functions.

Example 3.6.2. For the cognitive training example the GPPM becomes

$$\tilde{m}([t, \tilde{x_i}]; [b, c, d]) = c + d\tilde{x_i} + bt$$
$$\tilde{k}([s, \tilde{x_i}], [t, \tilde{x_i}]; [\sigma_\epsilon^2]) = k(s, t; \sigma_\epsilon^2) = \delta(s - t)\sigma_\epsilon^2.$$

3.6.2. Introduction to Unobserved Heterogeneity

Unobserved heterogeneity refers to inter-individual variation that is not explained by IVs. In psychology, the most popular approach to model unobserved heterogeneity is the random effects approach. Let θ_p be a particular parameter and θ_{ip} its person-specific counterpart. The random effects approach is to assume that the person-specific parameters θ_{ip} vary across persons around a mean of the parameter with some variance. More generally, the random effects approach refers to imposing any between-person distribution on a parameter. Every person-specific parameter θ_{ip} is considered to be a realization of the random variable $\theta_p \sim \mathcal{N}(\mu_{\theta_p}, \sigma_{\theta_p}^2)$. Formally, this can be equivalently expressed by assuming that every person-specific parameter is a random variable with the distribution $\theta_{ip} \sim \mathcal{N}(\mu_{\theta_p}, \sigma_{\theta_p}^2)$. Hence, the random effects approach effectively changes some parameters from fixed to random variables. Both the mean μ_{θ_p} and the variance $\sigma_{\theta_p}^2$ of the random variable θ_p are typically estimated and thus become parameters of the statistical model. If multiple parameters are converted to random variables, the covariances between them also have to be specified or estimated.

Example 3.6.3. Returning to the cognitive training example, a reasonable assumption would be to assume that the starting point a as well as the learning rate b vary between persons. For implementation, one can assume that each person-specific vector $[a_i, b_i]^\top$ has the joint distribution

$$\begin{bmatrix} a_i \\ b_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix} \right).$$

In comparison to observed heterogeneity the specification of unobserved heterogeneity is more limited for GPPMs. There are many parameters for which a between-person variation cannot be specified without violating the central assumption of GPPM that each persons' data are a realization of a GP. This restriction is not unique to GPPM. The types of random effects that are employed in hierarchical linear modeling (HLM) and SEM can all be specified using GPPM.

In the remainder of Section 3.6, I will describe for which kinds of parameters random effects can be specified. Essentially, random effects can be specified for linear parameters of the mean function. I will also show how the approach to modeling observed heterogeneity and the random effects approach can be combined. Specification of random effect leads to a violation of the assumptions that a persons data are a realization of a GP for most parameters that are not linear parameters of the mean function.

3.6.3. Implementation of Unobserved Heterogeneity

One important family of parameters for which random effects can be specified without violating the assumption that a persons' data are a realization of a GP are linear parameters of the mean function. Let the mean function be of the form

$$m(x;\theta) = f(x;\theta_1)^\top \theta_2 + h(x;\theta_3), \tag{3.6.1}$$

where the parameter vector $\theta = [\theta_1, \theta_2, \theta_3]$ is partitioned into parameters $\theta_1, \theta_2, \theta_3$. $f(\theta_1)$ is a vector-valued, and $h(\theta_3)$ a scalar-valued function. Using the random effects approach, unobserved heterogeneity is introduced by individualizing the parameter θ_2 and assuming that for each person the corresponding individualized parameter θ_{i2} has the distribution $\mathcal{N}(\mu_{\theta_2}, \Sigma_{\theta_2})$. In this way, the mean function itself becomes a GPPM. To see this, first note that for every value x of the IVs the corresponding value of the mean function $m(x;\theta_i)$ is the result of a linear transformation of the Gaussian random vector θ_{i2} and, thus, a Gaussian random variable. The mean of this random variable is

$$\begin{aligned}\mathbb{E}[m(x;\theta_i)] &= \mathbb{E}[f(x;\theta_1)^\top \theta_{i2} + h(x;\theta_3)] = \mathbb{E}[f(x;\theta_1)^\top \theta_{i2}] + h(x;\theta_3) \\ &= f((x;\theta_1)^\top \mu_{\theta_{i2}} + h(x;\theta_3).\end{aligned}$$

The covariance between the two random variables $m(x;\theta_i)$ and $m(x';\theta_i)$ is as follows. Note that, for every x, $m(x;\theta_i)$ can be interpreted as a random variable that is obtained by linearly transforming the random vector θ_{i2} with the matrix $f(x;\theta_1)^\top$. Thus, the covariance matrix Σ_m of the random vector $[m(x;\theta_i), m(x';\theta_i)]$ can be obtained through $\Sigma_m = A\Sigma_{\theta_2}A^\top$ with

$$A = \begin{bmatrix} f(x;\theta)^\top \\ f(x';\theta)^\top \end{bmatrix}.$$

The covariance between the random variables $m(x;\theta_i)$ and $m(x';\theta_i)$ amounts to the only non-diagonal entry (i.e., to the entry in the second row and first column, or equivalently in the first row and second column) of Σ_m, which is $f(x;\theta_1)^\top \Sigma_{\theta_2} f(x';\theta_1)$.

Since the mean is now a GPPM, the fact that the sum of two GPPMs is a GPPM again (see Theorem 3.5.1) can be used to obtain the GPPM that implements the random effects approach. Let $k(x, x';\theta)$ be the original covariance function. The new mean $\tilde{m}$ and covariance function $\tilde{k}$ function are as follows

$$\tilde{m}(x;\theta) = f(x;\theta_1)\mu_{\theta_2} + h(x;\theta_3) \tag{3.6.2}$$

$$\tilde{k}(x, x';\theta) = k(x, x';\theta) + f(x;\theta_1)^\top \Sigma_{\theta_2} f(x';\theta_1) \tag{3.6.3}$$

The mean vector μ_{θ_1} and the covariance matrix Σ_{θ_2} of θ_2 have become parameters.

Example 3.6.4. For the cognitive training example we wanted to allow the starting ability a, as well as the learning rate b to vary between persons. Thus, the model

changes to

$$\begin{aligned} m(t;[a_i,b]) &= a_i + b_i t \\ k(s,t;\sigma_\epsilon^2) &= \delta(s-t)\sigma_\epsilon^2 \end{aligned} \quad \text{with} \quad \begin{bmatrix} a_i \\ b_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix} \right)$$

We identify $f(x;\theta_1)^\top = [1,t]$, with $x = t$, and $\theta_1 = \{\}$, $\theta_{i2} = [a_i, b_i]$, $\mu_{\theta_2} = [\mu_a, \mu_b]^\top$,

$$\Sigma_{\theta_2} = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix},$$

$h(x;\theta_3) = 0$, and $k(x,x';\theta) = \delta(s-t)\sigma_\epsilon^2$, with $x = s$, $x' = t$. By inserting in Equations 3.6.2 and 3.6.3, the corresponding random effects GPPM can be derived:

$$\tilde{m}(t;\theta) = \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix} = \mu_a + \mu_b t \tag{3.6.4}$$

$$\begin{aligned} \tilde{k}(s,t;\theta) &= \delta(s-t)\sigma_\epsilon^2 + \begin{bmatrix} 1 & s \end{bmatrix} \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix} \begin{bmatrix} 1 \\ t \end{bmatrix} \\ &= \delta(s-t)\sigma_\epsilon^2 + \sigma_a^2 + \sigma_{ab}(s+t) + s\sigma_b^2 t. \end{aligned} \tag{3.6.5}$$

This is the GPPM representation of the LGCM (see Section 3.2.4).

3.6.4. Mixing Observed and Unobserved Heterogeneity

Mixing unobserved and observed heterogeneity is common in hierarchical models where between-person differences are partly unobservable but one also attempts to be explain them by available IVs. It is also possible using GPPM. First, the parameters θ_1, θ_3 for $f(x;\theta_1)$ and $h(x;\theta_3)$ can be made person-specific via a deterministic function $f(\theta, \tilde{x}_i, \tilde{\theta})$. Second, the mean vector μ_{θ_2} and the covariance matrix Σ_{θ_2} are parameters of a GPPM. Thus, they can also be made person-specific using a deterministic function.

Example 3.6.5. The effect of parents' income on the starting point a of the cognitive training study can be included in the random effects GPPM previously developed as follows: $\mu_{ia} = \mu_0 + c\tilde{x}_i$, where μ_0 is now the expected mean at parents' income of zero. Through substituting μ_a by μ_{ia} in Equation 3.6.4, the new mean function is obtained:

$$\tilde{m}([t,\tilde{x}_i];\theta) = \mu_0 + c\tilde{x}_i + \mu_b t$$

The covariance function remains unchanged.

3.6.5. Limitations for Unobserved Heterogeneity

While allowing for random effects on nonlinear parameters of the mean function or parameters of the covariance function might be conceptually desirable to model unobserved heterogeneity in those parameters, this poses a technical problem.

A nonlinear mean parameter is any parameter that cannot be written in the form as θ_2 in Equation 3.6.1. For nonlinear mean parameters, the technical problem is that for many common nonlinear transformations, the resulting random vector is not Gaussian. For example, a standard normal random variable Y, results in Y^2 being Chi-squared and $\exp(Y)$ log normally distributed. As a consequence, imposing a random effect on a nonlinear parameter of the mean function typically leads to the mean function becoming a parameterized stochastic process that is not a GP. Thus, within the definition of GPPM used in this work, random effects are not possible for many nonlinear parameters of the mean function.

Example 3.6.6. An exponential growth function for the mean might be a more realistic assumption in the training study. This changes the mean function to

$$m(t; a, b) = a(1 - e^{-bt}).$$

$a \in \mathbb{R}$ is now the theoretical ability level that is reached with unlimited training time and $b \in R_0^+$ characterizes the learning speed. The bigger b is, the faster learning occurs. As in the case of the linear learning model, it is probably a good assumption to allow for unobserved heterogeneity in the form of random effects for the maximally achievable ability level a as well as the learning speed b. Within GPPM, this is possible for the ability level a but not for the learning speed b. If it is assumed that b_i is a random variable that is distributed according to a Gaussian distribution, it follows that the mean function is a random variable that is distributed according to a log-normal distribution. Thus, the resulting model cannot be described as GPPM.

Similarly, reasonable between-person distributions on any parameter in the covariance function will likely lead to a violation of the GP assumption. That is, the statistical model becomes a set of non-GP stochastic processes. Investigating which combinations of between-person distributions and covariance function parameters do not violate the GP assumption needs to be examined in the future. However, the conjecture is that no reasonable between-person distribution on any parameter of the covariance function complies with the GP assumption. For present purposes, I will only show that imposing a uniform distribution on the constant covariance function $k(s, t) = \sigma^2$, which imposes a constant variance σ^2 for every time point, violates the GP assumption.

The Gaussian distribution might be the first choice for the between-person distribution of a variance parameter. However, the variance parameter may not be positive. This cannot be implemented using the Gaussian distribution. One easy approach to ensure that only positive values are possible is to use the uniform distribution over some positive range.

Let $m(x; \theta)$ be any mean function and the covariance function be $k(s, t) = \sigma^2$. If a uniform distribution $[0, c]$ over the variance parameter σ^2 is imposed, the model-implied

distribution for any value x of the IVs becomes

$$\begin{aligned} p(y|x) &= \int_0^\infty \mathcal{N}(y; m(x;\theta), \sigma^2) p(\sigma^2) \mathrm{d}\sigma^2 \\ &= \int_0^c \mathcal{N}(y; m(x;\theta), \sigma^2) \frac{1}{c} \mathrm{d}\sigma^2 \\ &= \frac{1}{c} \underbrace{\int_0^c \mathcal{N}(y; m(x;\theta), \sigma^2) \mathrm{d}\sigma^2}_{\text{not a Gaussian density}} . \end{aligned} \tag{3.6.6}$$

As a consequence, the statistical model is no longer a GPPM.

Whether random effects on nonlinear parameters of the mean function or on parameters of the covariance function are necessary for the modeling situations that typically occur in psychology needs to be investigated. Classical psychological analysis methods like SEM also exclude random effects on these parameters. Thus, typical psychological analyses are performed without random effects on these parameters. At the same time, it is straightforward to come up with convincing examples where random effects on these parameters intuitively seem reasonable, if not even necessary (see Example 3.6.6).

Extending GPPM to allow for random effects on any parameter seems relatively difficult, as the core assumption that every persons' data are a realization of a GP is violated. However, it should be possible.

3.7. Statistical Inference for Gaussian Process Panel Models

With GPR stemming from the Bayesian inference tradition, performing Bayesian inference for GPPMs is the natural choice. Bayesian inference for GPPMs is conceptually straightforward. Given a data set $D = (X, \mathbf{y})$, with $X = (X_1, \ldots, X_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$, GPPMs

$$Y_i(x) \sim \{\mathcal{GP}(m(x), k(x, x')) : \theta \in \Theta\}$$

reduce to regular statistical models of the form

$$p(\mathbf{y}|X) \in \left\{ \prod_{i=1}^N \mathcal{N}(y_i, m(X_i;\theta), k(X_i, X_i;\theta)) : \theta \in \Theta \right\}.$$

By imposing a prior $p(\theta)$ on the parameters θ, the posterior density $p(\theta|\mathbf{y}, X)$ can be calculated using Bayes' rule.

For many combinations of prior, mean, and covariance functions the posterior will not be a Gaussian distribution. Even worse, for many combinations an analytical solution for the posterior density does not exist. Thankfully, the Bayesian community has developed many solutions to this problem, which are general enough to also be applicable to GPPMs. Markov chain Monte Carlo methods (Diaconis, 2009) in particular can be used to perform Bayesian inference for any GPPM, prior combination.

While using Bayesian inference for GPPMs is possible, I will concentrate on developing frequentist inference procedures for GPPM. Researchers familiar with frequentist inference can then move from their known longitudinal modeling approaches (e.g., HLM or SEM) to GPPM more easily by just learning a new formal language to express their hypotheses about change trajectories and dynamics. This voids the need for familiarization with Bayesian inference.

As a reminder, the central forms of frequentist inference are point estimation, set estimation, and hypothesis testing. In the remainder of Section 3.7, I develop frequentist inference procedures for these three central forms of inference for GPPM. In Section 3.7.4, I will demonstrate how person-specific predictions can be obtained. I close this section with a recommendation regarding model selection and validation.

3.7.1. Point Estimation

I propose using the ML estimator as the point estimator for GPPMs. For a given data set $D = (X, \mathbf{y})$ the likelihood function for a GPPM is

$$L(\theta|D) = p(\mathbf{y}|X, \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i; M(X_i; \theta), K(X_i, X_i; \theta)).$$

The ML estimate is thus

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta|D).$$

For most GPPMs the maximum of the likelihood function can not be derived exactly. As a remedy, numerical optimization methods can be used. Any numerical optimization method that only relies on repeated function evaluations can be employed. For details about the optimization algorithm used in my implementation of ML estimation for GPPMs, see Section 3.8.2.

3.7.2. Hypothesis Testing

As the hypothesis test for GPPMs, I propose using the likelihood-ratio test, which is also used for SEMs. Remember that one important condition for the likelihood-ratio test is that the null hypothesis can be expressed by a constraint $g(\theta) = 0$. A GPPM can be restricted in such a way. For the likelihood-ratio test to be valid, i.e., for the test statistic to converge to a Chi-squared distribution both the restricted ML estimator $\hat{\theta}_R(Y)$ and the unrestricted estimator $\hat{\theta}(Y)$ need to be asymptotically Gaussian (Taboga, 2012c).

A complete proof of the asymptotical normality of the estimators is not within the scope of this text. However, I will provide a sketch of a proof: I assume that the number of time points observed for each person is equal. It is known that the ML estimator and the restricted ML estimator are asymptotically Gaussian for SEMs. Both a SEM and the statistical model implied by GPPM for a particular data set can be written in the form

$$p(\mathbf{y}|X) \in \left\{ \prod_{i=1}^{N} \mathcal{N}(y_i; \mu(X_i; \theta), \Sigma(X_i; \theta) : \theta \in \Theta \right\}.$$

For a SEM the implied mean $\mu(X_i;\theta)$ and covariance matrix $\Sigma(X_i;\theta)$ are the same for every person. Furthermore, the mean $\mu(X_i;\theta)$ and the covariance matrix $\Sigma(X_i;\theta)$ are restricted to be of the forms denoted in Definition 3.2.1. In contrast to that, the implied mean $\mu(X_i;\theta)$ and covariance matrix $\Sigma(X_i;\theta)$ for GPPMs may be different for every person and can in principle have any form (as long as $\Sigma(X_i;\theta)$ is always a valid covariance matrix).

Thus, GPPM extends SEM in two ways: first, there is no restriction on the parameterization of the mean vector and the covariance matrix, and second, the mean and the covariance matrix may be different for every person. The first extension does not violate any of the assumptions for asymptotic normality (Taboga, 2012d). The second extension violates the iid assumption. However, since the person-specific differences in the mean and the covariance matrix are produced by entering the IVs into template mean and covariance matrices, the conditional iid assumption still holds. The conditional iid assumption is sufficient for the restricted as well as the unrestricted (conditional) ML estimator to be Gaussian.

3.7.3. Confidence Regions

As is the case for SEMs, likelihood-based confidence regions can be used for GPPMs. This is done by applying the likelihood-ratio test to generate confidence regions. The rationale is identical to the one presented for SEM in Section 3.2.2. For convenience, I briefly repeat it here.

Let $\{\theta_1,\ldots,\theta_P\}$ be the parameters of a GPPM. To obtain a confidence region for an arbitrary subset of parameters $\theta_s \subset \{\theta_1,\ldots,\theta_P\}$, the following strategy can be used. Let Θ_s be the subspace of the parameter space Θ that corresponds to the parameters of interest and $\theta_s \in \Theta_s$ a particular parameter value. Let $\varphi_{\theta_s}(y)$ be the likelihood-ratio test for the hypothesis that the true value θ_s^* for the parameters is θ_s with the size α, then a confidence set estimator $\delta(y)$ with the confidence level $1-\alpha$ for the parameters θ_s can be obtained as follows:

$$\delta(y) = \{\theta_s \in \Theta_s : \varphi_{\theta_s}(y) = 0\}.$$

Thus, the resulting confidence set contains all parameter values $\theta_s \in \Theta_s$ for which the corresponding hypothesis test $\varphi_{\theta_s}(y)$ did not reject the null hypothesis.

3.7.4. Person-Specific Prediction

Person-specific prediction refers to making predictions for the DV of a particular person i based on the corresponding IVs. Thus, in particular, it applies to extra- and interpolation over time for a person i.

When using GPPM, performing person-specific predictions comes naturally. The ML estimator returns a parameter $\hat{\theta}$, which represents a GP. The joint distribution implied by the GP for the random vector $Y_i(X_i) = [Y_i(x_{i1}),\ldots,Y_i(x_{iJ})]$, representing the observed time points as indexed by the IVs within x_{ij}, and the random vector $Y_i(X_i^*)$,

representing the unobserved time points as indexed by the IVs within x^*_{ij}, is

$$\begin{bmatrix} Y_i(X_i) \\ Y_i(X_i^*) \end{bmatrix} \Bigg| X_i, X_i^* \sim \mathcal{N}\left(M\left(\begin{bmatrix} X_i \\ X_i^* \end{bmatrix} \right), \begin{bmatrix} K(X_i, X_i) & K(X_i, X_i^*) \\ K(X_i^*, X_i) & K(X_i^*, X_i^*) \end{bmatrix} \right) \tag{3.7.1}$$

Because a realization of the random vector $Y_i(X_i)$ has been observed, it is natural to compute the conditional distribution of $Y_i(X_i^*)$ given the observation y_i for $Y_i(X_i)$. Note that this is a Bayesian argument. In parallel to the predictive distribution in GPR, the conditional distribution of the unobserved time points $Y_i(X_i^*)$ given the observation y_i is

$$\begin{aligned} Y_i(X_i^*)|X_i, X_i^*, y_i &\sim \mathcal{N}(\mathbb{E}(Y_i(X_i^*)|X_i, X_i^*, y_i), \mathrm{Cov}(Y_i(X_i^*)|X_i, X_i^*, y_i)), \text{ with} \\ \mathbb{E}(Y_i(X_i^*)|X_i, X_i^*) &= M(X_i^*) + K(X_i^*, X_i)[K(X_i, X_i)]^{-1}(y_i - M(X_i)) \\ \mathrm{Cov}(Y_i(X_i^*)|X_i, X_i^*) &= K(X_i^*, X_i^*) - K(X_i^*, X_i)[K(X_i, X_i)]^{-1}K(X_i, X_i^*). \end{aligned}$$

If point estimates for the unobserved time points represented by $Y_i(X_i)$ are required, the MAP estimate or any other Bayesian point estimation technique can be used.

3.7.5. Model Selection and Validation

Remember model selection refers to selection of a model out of a set of candidate models, whereas model validation refers to assessment whether a model is correctly specified.

Since GPPM is similar to SEM, one can adapt the model selection and validation procedures used for SEM. These can be categorized into the likelihood-ratio test against a saturated model and penalty-based fit indices (see Section 3.2.3). Adapting these methods for model selection is relatively straightforward. I have shown that the likelihood-ratio test can be used for GPPMs in Section 3.7.2. Thus, the likelihood-ratio test can also be used to select between two GPPMs. Fit indices as used in SEM can also be used for selecting between GPPMs. I have implemented both the Akaike information criterion (AIC) and the BIC for GPPMs. In Section 4.2, I will demonstrate how the likelihood-ratio test and these fit indices can be used for selecting between GPPMs.

Adapting the methods used in SEM for model validation is less straightforward. For the likelihood-ratio test it is not obvious what the saturated model corresponding to a given GPPM should be. Since SEMs describe a statistical model for a random vector, the saturated model is "all Gaussian *distributions*." GPPMs describe a model for a stochastic process. Thus, the natural saturated model is "all Gaussian *processes*." Whether this is the correct translation and how the ML can be obtained under the saturated GPPM remains to be investigated.

To validate GPPMs, fit indices like the AIC and the BCI together with their recommended thresholds can, in principle, be used. However, it is not obvious that the thresholds developed for SEMs are also valid for GPPMs. I will not investigate these issues here for reasons of space. Instead, I will introduce cross-validation for the selection and validation of GPPMs in this section. My motivation to use cross-validation is as follows:

I have mentioned several concerns about model validation and selection practices in SEM before (see Section 3.2.3). Essentially, neither the likelihood-ratio test nor the proposed fit indices are suitable to validate the assumption that a model is correctly specified.

The likelihood-ratio test is the only procedure that tries to test the assumption that the model is correctly specified. Besides being seriously flawed (see Section 3.2.3), the likelihood-ratio test for the null hypothesis, "the model is correctly specified," almost always rejects it, providing strong evidence for the hypothesis that in practice, models will almost always be misspecified.

The penalty-based fit indices used for SEM acknowledge that every model will be misspecified and measure how well a given model approximates the reality instead (see also Little (2013, p. 108).

Like the fit indices, cross-validation does not directly assess whether a model is correctly specified. It rather estimates the validity of a given model in terms of how well it predicts data points that were not included in the data set D that was used for statistical inference. Many fit indices, for example the AIC, have actually been motivated as instruments to estimate the out-of-sample predictive performance solely on the basis of the data set D. Thus, the results obtained by cross-validation should be relatively similar to the ones obtained by the AIC approach. Which measure to choose for model selection and validation might depend on the family of models employed and the goal of the data analysis. However, I agree with Kearns, Mansour, Ng, and Ron (1997), who have compared many model selection methods: because of the favorable properties and the generality of cross-validation the burden of proof that a given penalty-based method performs better for a given model and data class combination lies with the practitioner who favors penalty-based procedures.

In the remainder of this section, I will discuss how to apply cross-validation to GPPMs. I have already introduced cross-validation for learning algorithms that return a prediction function, but GPPM does not contain a learning algorithm that returns a prediction function. However, the combination of a GPPM and the ML estimator can be interpreted as learning algorithm. Instead of a prediction function, the learning algorithm returns a GP. The GP could be transferred into a conventional prediction function by simply using its mean as the prediction. This would, however, ignore the covariance structure. Therefore, I propose evaluating the GP directly. To do this, I employ a utility function in parallel to the concept of a loss function. For every person y_i, the utility function is simply the log-likelihood of this person's data under the GP. Thus, for a data set D of N persons with corresponding partition P, the cross-validated log-likelihood of a GPPM, as represented by its corresponding parameterized mean m and covariance function k and parameter space Θ, is

$$R(m, k, \Theta) = \frac{1}{N} \sum_{D_l \in P} \sum_{(X_i, y_i) \in D_l} \log \left(\mathcal{N} \left(y_i; M \left[X_i; \hat{\theta} \left(D \setminus D_l \right) \right], K \left[X_i, X_i; \hat{\theta} \left(D \setminus D_l \right) \right] \right) \right).$$

$D \setminus D_k$ denotes the set difference. That is, the set $D \setminus D_l$ contains the elements, in this

case persons, that are in D but not in D_l. $\hat{\theta}(D \setminus D_l)$ describes the ML estimate of the parameters based on the data set $D \setminus D_l$. Assuming that the data sets $D \setminus D_l$ are of the same size, the cross-validated likelihood describes the expected log-likelihood of the data of a new person under the GP that is obtained by fitting a GPPM with a data set of the size $|D \setminus D_l|$ using ML. Thus, selecting the GPPM with the highest cross-validated log-likelihood selects the GPPM that leads to the GP best explaining the reality.

For model validation the cross-validated log-likelihood needs to be accessed in absolute quantities. One approach to achieve this is to compare the cross-validated log-likelihood of the proposed model to the cross-validated log-likelihood of other models.

In the remainder of this text, I will report the *negative* cross-validated log-likelihood. I do this since for fit indices, like the BCI and the AIC, a lower value usually refers to a better model.

3.8. Implementation of Gaussian Process Panel Modeling

In this section, I will explain how I implemented the core procedures presented in the preceding sections. As the basis for the implementation, I used the GPR toolbox Gaussian processes for machine learning (GPML) (Rasmussen & Nickisch, 2010) implemented in MATLAB.

3.8.1. Model Specification

The specification of a GPPM

$$Y_i(x) \sim \{\mathcal{GP}(m(x), k(x, x') : \theta \in \Theta)\}$$

is formally identical to the specification of the set of statistical models

$$f(x) \sim \{\mathcal{GP}(m(x), k(x, x') : \theta \in \Theta)\}$$

as used for model selection in GPR. Both require parameterized mean and covariance functions. Thus, the existing procedure to specify parameterized mean and covariance functions can be reused as the model specification procedure for GPPM.

3.8.2. Maximum Likelihood Estimation

If the number of persons $N = 1$ in a GPPM, the ML estimate $\hat{\theta}_{\mathrm{ML}}$ for the statistical model

$$Y_i(x) \sim \{\mathcal{GP}(m(x), k(x, x') : \theta \in \Theta)\}$$

is identical to the model $\hat{\theta}$ that maximizes the model evidence out of the set of models

$$f(x) \sim \{\mathcal{GP}(m(x), k(x, x') : \theta \in \Theta)\}$$

in the context of GPR. Thus, the algorithm for model selection via model evidence maximization could be reused to obtain the ML estimate. However, in the language of

GPPM, conventional GPR applications only deal with $N = 1$ data sets. Thus, I had to extend the model selection algorithm to allow for an arbitrary number of persons.

The available model selection algorithm works for one time series y_i and corresponding IVs X_i. It maximizes the likelihood function

$$p(y_i|X_i, \theta) = \mathcal{N}(y_i; M(X_i; \theta), K(X_i, X_i; \theta))$$

with respect to θ. For arbitrary GPPMs an algorithm that maximizes the likelihood function

$$p(\mathbf{y}|X, \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i; M(X_i), K(X_i, X_i))$$

with respect to the parameters θ is needed.

To find the parameter value $\hat{\theta}$ that maximizes the likelihood function, the existing algorithm minimizes the negative log-likelihood $-\log(p(y_i|X_i, \theta))$. To find the minimum of the negative log-likelihood, a gradient descent algorithm is employed. The employed gradient descent algorithm can in principle find the (local) minimum of any function $f(\theta)$. To do this it requires a helper function that returns the value of the function $f(\theta)$ for any parameter value θ and the corresponding gradient $\frac{\partial f(\theta)}{\partial \theta}$. The gradient $\frac{\partial f(\theta)}{\partial \theta}$ is defined as follows

Definition 3.8.1. For any differentiable function $f : \mathbb{R}^P \to \mathbb{R}$, the gradient is the vector of partial derivatives:

$$\frac{\partial f(\theta)}{\partial \theta} := \left[\frac{\partial f(\theta)}{\partial \theta_1}, \ldots, \frac{\partial f(\theta)}{\partial \theta_P}\right]^\top$$

Thus, to extend the existing algorithm to allow for an arbitrary number of people, a function that computes the joint negative log-likelihood for everyone $-\log(p(\mathbf{y}|X, \theta))$ and its gradient $\frac{\partial -\log(p(\mathbf{y}|X,\theta))}{\partial \theta}$ was required. The function that computes the negative log-likelihood $-\log(p(y_i|X_i, \theta))$ for one person and its gradient $\frac{\partial -\log(p(y_i|X_i,\theta))}{\partial \theta}$ already existed. The joint negative log-likelihood for all persons is

$$-\log(p(\mathbf{y}|X, \theta)) = -\log\left(\prod_{i=1}^{N} p(y_i|X_i, \theta))\right) = \sum_{i=1}^{N} -\log(p(y_i|X_i, \theta))$$

Similarly, the gradient of the joint negative log-likelihood for all persons is

$$\frac{\partial \log(-p(\mathbf{y}|X, \theta))}{\partial \theta} = \frac{\partial \sum_{i=1}^{N} -\log(p(y_i|X_i, \theta))}{\partial \theta} = \sum_{i=1}^{N} \frac{\partial -\log(p(y_i|X_i, \theta))}{\partial \theta}.$$

Thus, the required function that returns the value as well as the gradient of the joint negative log-likelihood for all persons merely consists of a summation of the person-level negative log-likelihoods and its gradients.

As a consequence of using a gradient descent algorithm for ML estimation, only mean and covariance functions that can be differentiated at least once can be used. The likelihood function $p(y_i|X_i, \theta)$ is as often differentiable with respect to the parameters θ as $M(X_i; \theta)$ and $K(X_i, X_i; \theta)$ are differentiable.

In principle neither $M(X_i; \theta)$ nor $K(X_i, X_i; \theta)$ have to be (once) differentiable. For example $m(x; \theta) = k(x, x'; \theta) = |\theta|$, with $\theta \in \mathbb{R}$, are both in principle valid parameterizations but are not differentiable for $\theta = 0$ and any x, x'. However, in practice most used parameterized mean and covariance functions do lead to matrices $M(X_i; \theta)$ and $K(X_i, X_i; \theta)$ that are differentiable at least once.

3.8.3. Hypothesis Testing

The likelihood-ratio test is not available in the GPML toolbox so I implemented it to allow frequentist hypothesis testing in GPPM.

Computing the likelihood-ratio test requires two central values: The unconstrained ML estimate $\hat{\theta}$, and the restricted ML estimate $\hat{\theta}_R$, i.e., the ML estimate under the constraint $g(\theta) = 0$. Together with the degrees of freedom difference, which is a property of $g(\theta)$, these two values determine if the hypothesis $g(\theta) = 0$ is rejected or not.

I already developed the functionality for obtaining the unconstrained ML estimate for the sake of point estimation (see Section 3.8.2). The algorithm for obtaining the ML estimate under the constraint $g(\theta) = 0$ requires a constraint optimization algorithm, which accepts functional constraints. The GPML toolbox does not provide such a general constraint optimization algorithm since it is not needed for GPR. Thus, implementing the likelihood-ratio test in its general form requires substantial changes to the GPML toolbox.

For this purpose, I aimed at providing an algorithm that is able to find the constrained ML estimate for constraints of the form $g(\theta) = [\theta_{p_1} - c_1, \ldots, \theta_{p_Q} - c_Q]$, with $p_1, \ldots, p_Q \in \{1, \ldots, P\}$ and $c_1, \ldots, c_Q \in \mathbb{R}$, that is, constraints that set $Q < P$ parameters to a fixed value. There were two reasons for concentrating on this special case of the more general constraint class $g(\theta) = 0$. First, many hypotheses used in practice are of this form; second, this hypothesis class is sufficient to compute likelihood-based confidence regions.

For implementing this constraint class, I reused a model selection approach from the GPML toolbox that allows a specification of a prior over the hyper-parameters θ. In this approach a hyper-prior $p(\theta)$ is specified. Instead of selecting the hyper-parameter value that maximizes the model evidence $p(\mathbf{y}|X_i, \theta)$, the hyper-parameter value that maximizes the following term is selected: $p(\mathbf{y}|X_i, \theta)p(\theta)$. One form of hyper-prior supported by the GPML toolbox are so-called delta priors. A delta prior takes on the form

$$p(\theta) = \begin{cases} 1 & \text{if } \theta_p = c, \text{ for a } c \in \mathbb{R} \\ -\infty & \text{otherwise} \end{cases}.$$

The delta prior effectively constrains the optimization to $\theta_p = c$. GPML also allows the combination of multiple delta priors, which results in the following prior

$$p(\theta) = \begin{cases} 1 & \text{if } \theta_{p_1} = c_1 \text{ and } \theta_{p_2} = c_2 \text{ and } \ldots \ \theta_{p_Q} = c_Q \\ -\infty & \text{otherwise} \end{cases}.$$

As desired, this prior is equivalent to the constraint $g(\theta) = [\theta_{p_1} - c_1, \ldots, \theta_{p_Q} - c_Q]$.

3.9. Related Work

Related work can be placed into four categories: (1) Using GPR as nonlinear regression techniques for psychological data, (2) applying GPTSM to psychological data, (3) other applications of GPs in psychology, and (4) extending GPTSM for multiple time series.

Besides the work by Cox et al. (2012), to which I will return later, I was not able to find any publications within psychology that used GPR. However, GPR has been used in the related field of neuroimaging (Ashburner & Klöppel, 2011; Doyle et al., 2013; Friston et al., 2008; Hughes et al., 2014; Kaden, Anwander, & Knösche, 2008; Kauppi et al., 2015; Kostro et al., 2014; Macke, Gerwinn, White, Kaschube, & Bethge, 2011; Marquand et al., 2010; Marquand, Brammer, Williams, & Doyle, 2014; Ruigrok et al., 2014; Salimi-Khorshidi, Nichols, Smith, & Woolrich, 2011; Ziegler et al., 2014). The most popular application of GPR within neuroimaging uses it as a decoding method. In this context, decoding refers to using supervised learning to predict a DV on the basis of the IVs MRI data.

The work by Ziegler et al. (2014) poses an exception within neuroimaging. This group used GPTSM to generate a predictive distribution for the grey matter volume of a voxel based on biological age and other IVs. Although the data they employed is cross-sectional they model it as if it stemmed from one person, that is as a time series, and thus estimate a cross-sectional age gradient on volumetric changes in the brain. They allow for inter-individual variation by additionally using IVs beyond biological age.

Griffiths, Lucas, Williams, and Kalish (2009) use GPR quite differently. Instead of using it as a statistical method, they propose a process model for human function learning that is based on GPR.

I could also find a paper discussing the extension of GPTSM for the analysis of panel data. Within the field of statistics, Hall et al. (2008) propose using GPs to model panel data. They share my assumption that the data for each person y_i are considered an iid realization of the same GP

$$Y_i(x) \sim \mathcal{GP}(m^*(x), k^*(x, x')).$$

However, they estimate the mean and covariance functions differently. In this work, I propose basing inference on a set of GPs, which is represented by parameterized mean and covariance functions and estimating the parameters using ML. In contrast, they use nonparametric estimation techniques. They only discuss ways to obtain point estimates for the mean and covariance functions. In contrast to this work, they do not go

into obtaining set estimates or performing model selection. These differences might be explained by different foci of attention. Hall et al. (2008) seemed to be most interested in obtained person-specific predictions, whereas the interpretation of the estimates for the mean and covariance functions is also of interest in this work. Hall et al. (2008) also discuss how to include link functions to model non-Gaussian data such as count or binary data.

Within psychology, the work by Cox et al. (2012) is closest to this work. These authors have adapted GPR for trajectory analysis as used in cognitive psychology. This is applied to analyze a trajectory of a computer mouse used by a participant. A trajectory corresponds to a series of (t, x, y) triples, where t denotes time, and x, y the x and y position of the cursor. They model the x and y position as independent GPs. Thus, they treat the series of x and y positions as univariate time series. Experiments typically consist of multiple trials. A trajectory is observed in every trial . Thus, the data of one participant consist of multiple observed time series. As a consequence, it is formally identical to a panel data set. However, in contrast to panel data, the time series all originate form one person. To model multiple trials originating from one person Cox et al. (2012) suggest concatenating the data of all trials into one trial, that is, treating all trials from one person as large time series. To extend the method to trials from different conditions, they use the same approach as taking in this work to extend GPTSM to GPPM. The concatenated trials from each condition are modeled as the iid realization of one GP that is the same across all conditions. They further extend their method to multiple persons, which I will not discuss here because it complicated and not relevant in the present context. Their method is different to the iid assumption used here and consequently distinguishes itself from virtually all panel modeling methods. Cox et al. (2012) suggest using the exponential squared covariance function without performing model selection, that is, they do not suggest any measures for model selection. For estimation, in contrast to this work, they use conventional Bayesian techniques as are typically applied for GPTSM. For person-specific predictions they use the same approach as the one advocated here.

Thus, my work is the first work to discuss how to adapt GPTSM for the modeling of psychological panel data. While there have been some efforts to extend GPTSM for the hierarchical modeling of multiple time series (Cox et al., 2012) or even panel data (Hall et al., 2008), my work is also the first work to introduce a complete panel modeling method based on GPTSM. Specifically, the discussion of what kind of inter-individual variation a GPPM can accommodate, the frequentist inference procedures proposed for GPPM, and the in-depth comparison with existing panel modeling approaches that will follow in the next chapter are unique to this work.

4. Advantages of Gaussian Process Panel Modeling

In the previous chapter, I introduced the new panel modeling method Gaussian process panel modeling (GPPM). In this chapter, I will examine its properties by comparing GPPM and two commonly used modeling methods, longitudinal SEM and multiple-subject state-space modeling (SSM). This direct comparison will reveal multiple advantages that GPPM has vis-à-vis to the existing methods.

Among the practical advantages of GPPM are its ability to express a larger space of models of practical relevance, the ease with which it is able to represent continuous-time models, and its built-in mechanism for performing person-specific predictions. I will demonstrate these and further advantages on two example data sets that I analyzed using GPPM. This also serves as a practical demonstration of GPPM.

The comparison of longitudinal SEM and GPPM will reveal that longitudinal SEMs can be regarded as a special case of GPPMs. Thus, every longitudinal SEM can be translated into an equivalent GPPM. This provides the opportunity to use GPPM software to obtain ML estimates for longitudinal SEMs. Initial pilot studies have suggested that GPPM software might be able to compute ML estimates faster than conventional SEM software. Indeed, I will show in this chapter that GPPM software is indeed typically faster for the same model. Depending on the model, this difference is of different magnitude.

4.1. Relationships to Conventional Longitudinal Panel Modeling Approaches

4.1.1. Longitudinal Structural Equation Modeling

To investigate the relationship of GPPM to conventional longitudinal analysis methods and its benefits in comparison to conventional methods, I compare it against longitudinal SEM. Arguably, longitudinal SEM and hierarchical linear modeling (HLM) are the most widely used panel methods, and HLM can be considered a special case of longitudinal SEM (Curran, 2003).

First, I will show that any SEM can be described as an equivalent GPPM. It follows that any longitudinal SEM can be described as an equivalent GPPM. I will not restrict the proof to ordinary SEMs, but rather prove it for the broader family of extended SEMs with definition variables.

Theorem 4.1.1. For every extended SEM with definition variables, there is an equivalent GPPM.

Proof. To prove this theorem, I will provide a recipe with which every SEM can be translated into an equivalent GPPM. Both modeling techniques derive the model for all persons by specifying a model for one of them and making the iid assumption. Thus, it suffices to show that every person-level SEM can be specified using GPPM.

SEMs imply a person-level model for the random vector Y_i representing the data of person i of the form

$$Y_i \sim \{\mathcal{N}(F(I-A(d_i;\theta))^{-1}m(d_i;\theta), F(I-A(d_i;\theta))^{-1}S(d_i;\theta)(I-A(d_i;\theta))^{-1^\top}F^\top) : \theta \in \Theta\}. \tag{4.1.1}$$

For the definition of all terms see Section 3.2.1.

GPPMs imply a person-level model for the random process $Y_i(x)$ of person i of the form

$$Y_i(x) \sim \{\mathcal{GP}(m(x;\theta), k(x,x';\theta)) : \theta \in \Theta\},$$

with $x, x' \in \mathcal{X}$.

Note that both SEM and GPPM use the symbol m. In SEM, this symbol refers to the mean vector of the disturbances, whereas it refers to the mean function in GPPM. To avoid confusion, I will denote the mean vector in SEM by $m^{(\text{sem})}$ in the following.

A central difference between SEMs and GPPMs is that a SEM represents a statistical model for a random vector, whereas a GPPM represents a statistical model for a stochastic process. However, for a particular panel data set $\{(X_i, y_i), i \in 1, \ldots, N\}$ a GPPM is translated into a statistical model for a random vector:

$$xY_i(X_i) \sim \{\mathcal{N}(M(X_i;\theta), K(X_i, X_i;\theta) : \theta \in \Theta\}. \tag{4.1.2}$$

Thus, to show that every SEM can be represented as a GPPM, it remains to be demonstrated that a parameterized mean $m(x;\theta)$ and covariance function $k(x;\theta)$, and corresponding matrix of IVs X_i can be chosen such that any statistical model represented by a SEM (Equation 4.1.1) is identical to the statistical model implied by the corresponding GPPM for a particular data set (Equation 4.1.2).

Parametrical statistical models are determined by the parameter space Θ as well as by the functions that map every parameter value within the parameter space to a distribution. In the cases of SEM and GPPM, the parameter values are mapped to a corresponding distribution by mapping them to a mean vector and a covariance matrix, which I will refer to as moments in the remainder. Neither SEM nor GPPM demand any restrictions regarding the form of the parameter space Θ. Thus, it suffices to show that a set of independent variables X_i, a mean function $m(x)$, and a covariance function $k(x, x')$ can be found such that the moments implied by the SEM and the corresponding

GPPM are the same for each person i and parameter value θ.

$$M(X_i;\theta) \stackrel{!}{=} F(I - A(d_i;\theta))_i^{-1(\text{sem})} \tag{4.1.3}$$

$$K(X_i, X_i;\theta) \stackrel{!}{=} F(I - A(d_i;\theta))^{-1} S(d_i;\theta)(I - A(d_i;\theta))^{-1^\top} F^\top \tag{4.1.4}$$

For didactic reasons, I will first show how to find a mean function $m(x;\theta)$ and corresponding values of IVs X_i for a sample model. As an example, I use a LGCM with three measurement occasions. The corresponding path diagram has already been shown in Figure 3.1. The full algebraic representation is as follows:

$$m^{(\text{sem})}(\theta) = \begin{matrix} Y_1 \\ Y_2 \\ Y_3 \\ a \\ b \end{matrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu_a \\ \mu_b \end{bmatrix}, \qquad A(d_i;\theta) = \begin{matrix} \\ Y_1 \\ Y_2 \\ Y_3 \\ a \\ b \end{matrix}\begin{matrix} \begin{matrix} Y_1 & Y_2 & Y_3 & a & b \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & t_{i1} \\ 0 & 0 & 0 & 1 & t_{i2} \\ 0 & 0 & 0 & 1 & t_{i3} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \tag{4.1.5}$$

$$S(\theta) = \begin{matrix} \\ Y_1 \\ Y_2 \\ Y_3 \\ a \\ b \end{matrix}\begin{matrix} \begin{matrix} Y_1 & Y_2 & Y_3 & a & b \end{matrix} \\ \begin{bmatrix} \sigma_\epsilon^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\epsilon^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\epsilon^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_a^2 & \sigma_{ab} \\ 0 & 0 & 0 & \sigma_{ab} & \sigma_b^2 \end{bmatrix} \end{matrix}, \qquad F = \begin{matrix} \\ Y_1 \\ Y_2 \\ Y_3 \end{matrix}\begin{matrix} \begin{matrix} Y_1 & Y_2 & Y_3 & a & b \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

The corresponding parameter is vector $\theta = [\mu_a, \mu_b, \sigma_\epsilon^2, \sigma_a^2, \sigma_b^2, \sigma_{ab}]$. Only the $A(d_i;\theta)$ matrix is individualized using the definition variables $d_i = [t_{i1}, t_{i2}, t_{i3}]$. Thus, by replacing d_i with $[t_{i1}, t_{i2}, t_{i3}]$ in Equation 4.1.3, the property of the desired mean function is

$$M(X_i;\theta) \stackrel{!}{=} F(I - A([t_{i1}, t_{i2}, t_{i3}];\theta]))^{-1} m^{(\text{sem})}.$$

It remains to be shown that a mean function $m(x;\theta)$ and a matrix of IVs X_i exist that leads to fulfillment of this condition. As matrix X_i, I choose

$$X_i = \begin{bmatrix} t_{i1} & t_{i2} & t_{i3} & 1 \\ t_{i1} & t_{i2} & t_{i3} & 2 \\ t_{i1} & t_{i2} & t_{i3} & 3 \end{bmatrix}.$$

The last column indexes the observed variables. In the remainder, I will denote the index by j. Constructing a mean function that produces the same model-implied mean using this matrix of IVs X_i as the LGCM is trivial. It simply is

$$m([t_{i1}, t_{i2}, t_{i3}, j];\theta) = (F(I - A([t_{i1}, t_{i2}, t_{i3}], \theta]))^{-1} m)_j,$$

where $(z)_j$ denotes the jth entry of the vector z.

The desired covariance function can be constructed on the same matrix X_i in similar fashion. I will not describe the construction but rather continue with the proof for the general case. In the general case $m^{(\text{sem})}(d_i;\theta)$, $A(d_i;\theta)$, $S(d_i;\theta)$ all depend on the definition variables d_i. First, I again use repetitions of the definition variables augmented with an index for the corresponding observed variable as matrix X_i for each individual:

$$X_i = \begin{bmatrix} d_i & 1 \\ \vdots & \vdots \\ d_i & T \end{bmatrix}.$$

T refers to the number of observed variables in the SEM. In the mean function I denote the index of the observed variable as j, and the index for the second observed variable as k in the covariance function.

The mean function

$$m([d_i, j];\theta) = (F(I - A(d_i;\theta))^{-1} m(d_i;\theta))_j$$

fulfills the desired property. Equivalently, the covariance function

$$k([d_i, j], [d_i, k];\theta) = (F(I - A(d_i;\theta))^{-1} S(d_i;\theta)(I - A(d_i;\theta))^{-1^\top} F^\top)_{jk}$$

fulfills the desired property. □

Thus, the family of extended SEMs is a subset of the family GPPMs. The modeling approaches may still differ in the types of inferences that are typically performed. However, inference for SEMs as well as GPPMs is performed using ML as point estimation technique, the likelihood-ratio test as hypothesis test, and likelihood-based confidence intervals. Thus, every standard analysis that can be performed using SEM can equally be performed using GPPM.

The interesting question now is if the reverse is also true: Is GPPM a special case of SEM, and, thus, would both method be equally expressive? I also first start answering this question by answering the question: Can every GPPM be expressed as an equivalent SEM? Strictly, this cannot be the case since a GPPM describes a statistical model for a stochastic process whereas a SEM describes a statistical model for a random vector.

The second important difference is that GPPM poses no restriction on the parameterization of the mean and the covariance function, whereas conventional SEM restricts the parameterization of the mean and covariance matrices. This restriction, however, is removed when considering extended SEM.

Indeed, with extended SEM and definition variables the statistical model implied by a GPPM for a particular data set can be described. For every data set $(X, \mathbf{y})$, a GPPM reduces to a statistical model for a random vector of the form

$$p(\mathbf{y}|X) \in \left\{ \prod_{i=1}^{N} \mathcal{N}(y_i, M(X_i;\theta), K(X_i, X_i;\theta)) : \theta \in \Theta \right\}.$$

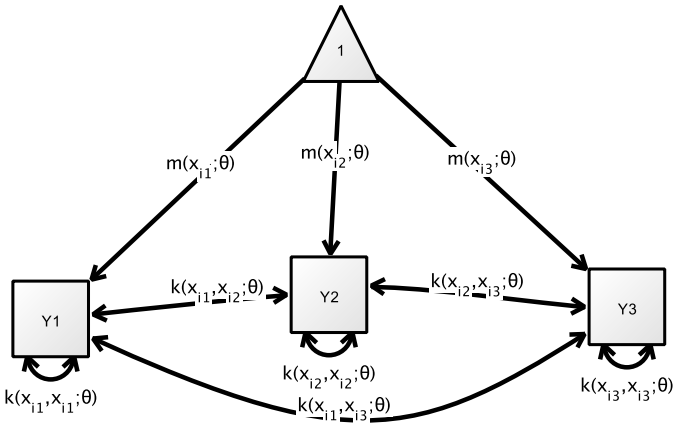


Figure 4.1.: Path diagram illustrating the translation of a GPPM into a SEM.

Thus, in parallel to the translation of a SEM into an equivalent GPPM one needs to find SEM matrices and definition variables such that

$$F(I - A(d_i;\theta))_i^{-1(\text{sem})} \stackrel{!}{=} M(X_i;\theta) \tag{4.1.6}$$

$$F(I - A(d_i;\theta))^{-1} S(d_i;\theta)(I - A(d_i;\theta))^{-1^\top} F^\top \stackrel{!}{=} K(X_i, X_I;\theta). \tag{4.1.7}$$

By restricting the set of candidate SEMs to those for which $A(d_i;\theta) = 0$, the conditions that need to be fulfilled simplify to

$$m^{\text{sem}}(d_i;\theta) = M(X_i;\theta) \quad S(d_i;\theta) = K(X_i, X_i;\theta)$$

If the number of observations is the same across all persons, this does not pose a problem. One simply equates the definition variables d_i and the IVs X_i, and defines the template matrices as

$$A(X_i;\theta) = 0 \quad m^{\text{sem}}(X_i;\theta) = M(X_i;\theta) \quad S(X_i;\theta) = K(X_i, X_i;\theta).$$

This can be done because extended SEM allows arbitrary complex functions of the definition variables and of the parameters for all entries of the SEM matrices. Assuming that the number of observations per person is 3, the GPPM translated from a SEM has a path diagram as displayed in Figure 4.1. However, even extended SEM does not allow the number of observations to differ between persons as is possible in GPPM. This problem can be solved by setting the number of observed variables to the maximum number of observations per person. For persons with fewer observations, the remaining observations are simply treated as missing values.

I have shown that every longitudinal SEM can be transformed into an equivalent GPPM. The statistical model implied by a GPPM can be transformed into an extended SEM. Using conventional SEM, this is not possible. The translation from SEM to

GPPM seems useful at least for some models. For a LGCM, for example, the GPPM representation provides a compact, alternative representation of the model (compare Equations 3.6.4 and 3.6.5 with Equation 3.2.9). In contrast, the translation of a GPPM into a SEM, as introduced in this section, hardly ever seems to be reasonable. The resulting SEM will necessarily be artificial and the graphical representation as offered by SEM will be cluttered. Most importantly, the SEM will not contain any linear regression, since the A matrix is 0. Thus, one might not even call this model a SEM but rather a generic covariance model. As a consequence, I will ignore the existence of this translation in the remainder. From a technical viewpoint, however, the translation is valuable. It shows that GPPM can be implemented by building on extended SEM software. The only major extension required is a function that translates a GPPM into an equivalent SEM.

The inference procedures proposed for GPPM in this work are not all commonly used for SEM. For point estimation, hypothesis testing, and confidence intervals, the procedures are identical, but I propose using cross-validation for model selection and validation, which is not common in the SEM community. Brandmaier, von Oertzen, McArdle, and Lindenberger (2013), however, do use cross-validation for SEMs.

Person-specific prediction (as presented in Section 3.7.4 for GPPM) is not commonly performed in longitudinal SEM. A recent book devoted to longitudinal SEM (Little, 2013) does not include anything about the topic. However, the problem of estimating person-specific factor scores, which has been discussed in the SEM literature (e.g., Estabrook & Neale, 2013), is closely related. Both problems consist of estimating a person-specific value based on the data for a person and a joint distribution for the data of all persons. Like the person-specific prediction method for GPPM, the expected posterior method to estimate factor scores (Estabrook & Neale, 2013) uses the conditional distribution of the factor score given the observed data to estimate the person-specific factor scores. Thus, although person-specific predictions are currently rarely used within longitudinal SEM, the expected posterior method to get factor scores may potentially be general enough to be reusable for obtaining person-specific predictions that would be identical to the GPPM-generated person-specific predictions.

Ultimately, SEM and GPPM differ in how they describe the model-implied moments, i.e., the model-implied mean and covariance matrices. In SEM, the model-implied moments of the observed variables are described using noisy linear equations. Thus, a SEM describes a statistical model for a Gaussian random vector. The moments of the random vector are implicitly described by the structural equations and have to calculated as shown in Equation 4.1.3 and Equation 4.1.4. Furthermore, not all parameterizations of the moments are possible; they are restricted to those that can be expressed via structural equations.

In GPPM, the model-implied moments of a GP are explicitly described via arbitrarily parameterized mean and covariance functions. Thus, there are four central differences compared to SEM. In GPPM:

1. The statistical model is described for a stochastic process rather than a random vector.

2. The model-implied moments are described explicitly, not implicitly.
3. Every parameterization of the model-implied moments is allowed not only those expressible by linear structural equations.
4. The concept of latent variables is not used explicitly.

These four, rather technical differences lead to a variety of practical differences. By describing the statistical model on a stochastic process, GPPM is well suited for continuous as well as discrete-time modeling. SEM, on the other hand is well suited for discrete-time modeling, but since a SEM is a statistical model for a random vector, continuous-time analyses are more cumbersome.

For readers not familiar with the concepts of discrete-time and continuous-time modeling, I provide a brief introduction. (for an in-depth discussion of continuous- versus discrete-time modeling in psychology, see, e.g., (Oud & Jansen, 2000; Oud & Singer, 2008; Voelkle et al., 2012)). In discrete-time modeling, variables that emerge over time are modeled at distinct time points, and for all time points in-between, no model is imposed. In contrast, continuous-time modeling treats the observed variables as discrete snapshots of an underlying continuous process. The continuous process is modeled directly rather than the discrete time points. Arguably, continuous-time modeling is better suited to model panel data, since the concept of interest is the continuous process and not a series of discrete snapshots. Besides this conceptual benefit, continuous-time modeling also has several practical advantages. Among the most important are: Different time intervals between as well as within persons are properly accounted for and inference results between studies are directly comparable.

Another advantage of the continuous-time perspective in combination with Bayesian inference as used for GPPM is that it makes person-specific prediction (i.e., inter- and extrapolations) straightforward. In the panel modeling context, person-specific predictions can most prominently be used to obtain person-specific trajectories that extend the observed time frame. For example, one could obtain a prediction for how the cognitive ability of a person develops as they age. Importantly, the prediction approach taken by GPPM does not only use the data for the respective person but also takes the data of all other persons into account. Additionally, GPPM does not only enable to estimate a most likely trajectory but also an uncertainty for the trajectory. As such, person-specific predictions as obtained by GPPM can be easily used as a screening device for interventions. To stay with the example, if the 95% credibility interval for the cognitive ability of a given person at a given age contains only values that are considerably worse than would normally be expected at this age, an intervention should be performed.

By specifying the model-implied moments directly rather than implicitly, ML estimates, and in turn likelihood-based confidence intervals can potentially be computed more efficiently for GPPMs. I will investigate this topic in depth in Section 4.3.

The direct specification of the model-implied moments also influences how a model is determined by a researcher. Describing the model-implied moments implicitly via structural equations and especially via path diagrams might be more straightforward than describing them explicitly. In GPPM, translating a theory into mean and covariance

functions potentially places a higher burden on the researcher. However, using the rule to compose the mean and covariance functions as the sum of simple mean and covariance functions established in Theorem 3.5.1, they can be constructed from a set of already available functions that represent important model classes.

A related topic is the non-existence of latent variables in GPPM. While not supporting latent variables directly, they can be included implicitly. As an example, review the specification of the LGCM in Equations 3.6.4 and 3.6.5. This approach allows inclusion of latent variables on the level of model specification, estimation, and hypothesis testing. However, person-specific predictions cannot be obtained for latent variables. To do this, a joint distribution of all measurements and latent variables is required. If an observed variable is simply a latent variable plus some measurement error, this can already be done using GPPM. For an example, see the person-specific predictions presented in Section 4.2.1. Extending GPPM to arbitrary combinations and roles of latent variables will be the subject of future work.

The fact that GPPM allows every parameterization means that a larger model space can be represented by GPPM. In other words, GPPM offers a variety of new ways to think about the temporal evolution of psychological processes. Indeed any model that is a set of GPs can be expressed. For example, a GPPM could be "any smooth process," for a given definition of smoothness. Using SEM this is not possible.

The model "any smooth process" as an example of a model that can be expressed in GPPM but not in SEM deserves a little more attention. It is a long-standing assumption in the natural sciences that "nature does not make jumps" ("natura non facit saltus"). This has been promoted by thinkers such as Leibniz (1704/1886) and Darwin (1859). The most straightforward mathematical implementation of this principle is continuity. Informally, a continuous function $f(x) = y$ can be drawn without removing the pen from the paper, i.e., there are no jumps in the function. The principle can also be extended to the derivative of a function. For example, not only the function describing the location of a car is continuous, but also the function describing its velocity. This can be further extended to the acceleration. Repeating this principle indefinitely leads to the requirement that the function $f(x) = y$ is infinitely often differentiable, and that all derivatives are continuous. A function satisfying this property is called smooth. Thus, it can be argued that if no more is known about a phenomenon that emerges over time the best assumption is smoothness.

A GP is not a function, it is a stochastic process. To extend the definition of smoothness to a stochastic process, one can look at the properties of the realization of a stochastic process (also known as sample path in the literature). Every realization of a GP can be described by a function $f(x) = y$. Thus, for every realization, properties of the resulting function such as continuity or smoothness can be examined. For the exponential squared covariance function

$$k(x, x'; [\sigma, l]) = \sigma^2 \exp\left(\frac{(x - x')^2}{-2l^2}\right) \quad \sigma, l \in \mathbb{R}^+$$

every realization is smooth. Thus, the exponential squared covariance function can be interpreted as a mathematical implementation of the "nature does not make jumps"

principle. In contrast to this, realizations of the popular LGCM and the continuous-time AR(1) model are continuous but not smooth.

Another more practical difference between SEM and GPPM is that they originate from quite different communities and as such come with different "default models." Faced with the problem of data analysis, many applied researchers working with panel data simply choose from a set of default longitudinal SEMs. Interestingly, while there is some overlap, the default models used for Gaussian process time series modeling (GPTSM) and longitudinal SEM differ although the problems for which they are used are similar. Therefore, GPPM extends the repertoire of default models from which applied researcher can choose. In this vein, I will compare the AR model as used in the SEM community to the similar exponential squared covariance function, which implements the "nature does not jump" assumption and is commonly used in the GPTSM community (cf. Section 4.2.1).

To summarize: every longitudinal SEM can be expressed as a GPPM. GPPM extends the space of expressible models. One example of a worthwhile model that can only be expressed using GPPM is the exponential squared model, which implements the "nature does not jump" assumption. Furthermore, GPPM is well suited for discrete-time and continuous-time modeling, whereas continuous-time modeling in SEM is cumbersome. The combination of continuous-time modeling and the deep rooting in Bayesian inference equips GPPM with a straightforward mechanism for providing person-specific predictions, which are not as easily obtained in longitudinal SEM. The model implied moments are described implicitly in SEM and explicitly in GPPM. Whether this difference leads to GPPM software being faster will be investigated in Section 4.3.

4.1.2. State-Space Modeling

Since one of the main differences between GPPM and SEM is that the former is able to express a statistical model for a stochastic process and is thus better suited for continuous-time modeling, it is obvious to compare GPPM against other techniques that are also able to do this. Within the social sciences, one of the most relevant methods is multiple-subject state-space modeling (SSM), as advocated by Boker (2007a, 2007b), Driver, Oud, and Voelkle (in press), Oud and Jansen (2000), Oud and Singer (2008), Voelkle et al. (2012).

I will only be able to briefly introduce multiple-subject SSM. For a more detailed introduction to the topic see Oud and Jansen (2000), Oud and Singer (2008), Voelkle et al. (2012).

Similarly to GPPM, multiple-subject SSM is originally a time series analysis approach that was extended to a panel modeling technique (Oud & Singer, 2008). Linear time-invariant Gaussian SSMs are of the form

$$\begin{aligned} \text{State equation:} \quad & \frac{\mathrm{d}L(t)}{\mathrm{d}t} = AL(t) + Bx(t) + \zeta(t) \\ \text{Output equation:} \quad & Y(t) = CL(t) + Dx(t) + \epsilon(t) \end{aligned}$$

$L(t)$ is a latent stochastic process. Its description, the state equation, is a so-called stochastic differential equation (SDE). A SDE can be seen as the continuous-time equivalent of a difference equation (Voelkle et al., 2012); for an elaborate introduction to SDEs, see Kuo (2006). The output equation encodes how the latent process $L(t)$ is mapped to the observable process $Y(t)$. A, B, C, D are matrices of appropriate size. $x(t)$ contains the values of some IVs at time point t. For every t, both $\zeta(t)$ and $\epsilon(t)$ are Gaussian random vectors with distribution $\zeta(t) \sim \mathcal{N}(0, Q)$ and $\epsilon(t) \sim \mathcal{N}(0, R)$.

A SSM is only fully specified by making assumptions about the value of the latent process $L(t_0)$ at some initial time point t_0. This value is an additional parameter of the model. Thus, the parameters of a SSM are $\theta = [A, B, C, D, Q, R, L(t_0)]$.

Given a value $L(t_0)$ of the latent process at an initial time point t_0, the solution of the state equation for all following time points $t > t_0$ is

$$L(t) = e^{A(t-t_0)}L(t_0) + \int_{t_0}^{t} e^{A(t-s)}Bx(s)ds + \int_{t_0}^{t} e^{A(t-s)}QdW(s). \tag{4.1.8}$$

I will now show that every SSM can be expressed as an equivalent GPPM.

Theorem 4.1.2. Every SSM can be expressed as an equivalent GPPM.

Proof: I will show that for every parameter value θ the solution for Equation 4.1.8 is a GP, and that consequently the set of stochastic processes for the observable process $Y(t)$ defined by any SSM is a set of GPs. It follows that every SSM can be expressed as an equivalent GPPM.

In the first term of Equation 4.1.8, $e^{A(t-t_0)}$ is a matrix exponential, the result of which is again a matrix. Thus, the first term is a vector.

The second term is an ordinary Riemann integral. However, the values of the IVs $x(s)$ are not known for all relevant time points $s \in [t_0, t]$. The values of the IVs $x(s)$ are only observed for a finite set of time points $[t_0, \ldots, t_T]$, which typically coincide with the time points for which a realization of observable process $Y(t)$ is obtained. To solve this problem, it is usually assumed that the IVs $x(s)$ do not change between two observations. It follows that for all time points s, such that $s_i \leq s < s_{i+1}$, $x(s) = x(s_i)$. Under this assumption, the Riemann integral can be computed and results in a vector.

The third term is a so-called Wiener integral (Kuo, 2006, Chapter 2.3). The solution of a Wiener integral $\int_a^b f(x)\mathrm{d}W(x)$ is a Gaussian random vector with a mean of 0 and the covariance matrix $\int_a^b f(x)f(x)^\top \mathrm{d}x$. Thus, the third term represents a Gaussian random vector with a mean of 0 and the covariance matrix

$$\mathrm{Cov}\left(\int_{t_0}^{t} e^{A(t-s)}QdW(s)\right) = \int_{t_0}^{t} e^{A(t-s)}Q(e^{A(t-s)}Q)^\top = \int_{t_0}^{t} e^{A(t-s)}QQ^\top e^{A^\top(t-s)}.$$

As a consequence, for every time point t, the latent process $L(t)$ is a Gaussian random vector. In other words, the latent process is a multivariate GP.

Since the observable process $Y(t)$ is the result of a linear mapping of the latent process $L(t)$, the observable process $Y(t)$ is also a GP. As to why this is the case, note that

for every time point t the value of the latent process $L(t)$ is a Gaussian random vector, and that the linear transformation of a Gaussian random vector results in a Gaussian random vector (see Theorem A.3.7 in Appendix A.3).

By parameterizing the model matrices $A, B, C, D, Q, R, L(t_0)$, a set of GPs, or in other words a GPPM, is described. □

The approach used to extend SSM to multiple subjects is the same as the one I used to extend GPTSM to GPPM. Each person's time series is considered an iid realization of the same GP as represented by a SSM. The modifications to allow for more forms of inter-individual variation added, for example, by Driver et al. (in press), Oud and Jansen (2000), Oud and Singer (2008), Voelkle et al. (2012), can all be considered special cases of the procedures to introduce inter-individual variation for GPPMs that I described in Section 3.6. Thus, the multiple-subject SSM can be considered a special case of GPPMs.

The reverse question, whether GPPMs can also be expressed as multiple-subject SSM, naturally arises. For some GPPMs, there is an equivalent multiple-subject SSM. However, there are GPPMs that can not be expressed as an equivalent multiple-subject SSM, and, thus, GPPM cover a truly larger model space than multiple-subject SSM. Consider, for example, the GPPM represented by the exponential squared covariance function

$$k(s,t) = \sigma^2 \exp\left(-\frac{(s-t)^2}{\rho}\right),$$

with $\rho > 0$. There is no SSM that can represent this covariance function (Hartikainen & Särkkä, 2010). However, for any GPTSM that only has time as the IV a SSM approximating it can be found (Hartikainen & Särkkä, 2010). Consequently, this probably pertains to any GPPM with only time as the IV.

The Kalman filter algorithm is commonly used in multiple-subject SSM for parameter estimation. The Kalman filter algorithm can be interpreted as an alternative way of computing the person-level likelihood. ML is also used as estimation method. The likelihood-ratio test is used as hypothesis test, and likelihood-based confidence intervals are the recommended approach to compute confidence intervals. Thus, GPPM can perform all those conventional analyses (parameter estimation, confidence intervals, hypothesis tests) that can be carried out with multiple-subject SSM.

To obtain person-specific predictions based on a multiple-subject SSM, the Kalman filter algorithm, which is used to compute person-level likelihoods, can also be used. This algorithm contains a prediction step, which computes the conditional distribution of the state $L(t)$ at time point t given all previous observations. The Kalman smoother algorithm can be used to obtain the conditional distribution of the state $L(t)$ given all other observations $p(L(t)|y_i)$. Arguably, Kalman filter predictions should only be used in psychology when they are identical to the Kalman smoother predictions, i.e., when only data preceding the time point of interest have been observed. Thus, performing person-specific predictions, as in GPPM, is implemented by computing the conditional distribution of the unknowns given all the knowns. In contrast to GPPM, the prediction is not about the value of the observable process $Y(t)$ but rather about the latent state process $L(t)$. In principle, GPPM can also be extended to allow for latent predictions. Indeed, in Section 4.2.1, I provide person-specific predictions for latent constructs.

However, GPPM currently does not support latent predictions in the general form supported by multiple-subject SSM. Predictions about the observable process $Y(t)$ as used in GPPM can also be obtained using multiple-subject SSM. Using the output equation, the prediction of the state of the latent process $L(t)$ as returned by the Kalman smoother can be mapped to a prediction for the observable process $Y(t)$.

GPPM and multiple-subject SSM mostly differ in how the model-implied moments are specified, in analogy to the difference I noted earlier between longitudinal SEM and GPPM. However, in contrast to longitudinal SEM, both GPPM and multiple-subject SSM describe the statistical model as a set of GPs. Thus, they are both well suited for continuous-time modeling. In GPPM, the set of GPs is specified explicitly by a parameterized mean and covariance functions, whereas it is defined implicitly by a linear Gaussian SDE in multiple-subject SSM. Therefore, any set of GPs can by described in GPPM, whereas only those sets that can by described by a parameterized linear Gaussian SDE are possible in multiple-subject SSM.

The form of a researcher's theory determines whether translating it into a mean and covariance functions representation is easier than translating it into a SDE. If the theory is about the way a system changes from one time point to another, as is the case for a AR(1) model, deriving the corresponding multiple-subject SSM should be the easier approach. Models that focus on describing the change of a system from one time point to another are referred to as dynamic models (Hertzog & Nesselroade, 2003; Voelkle, in press). If the theory on the other hand comprises a complete analytic description of the to-be-expected change, which is usually represented as a function of some representation of time, as is the case for LGCMs, GPPM seems to be better suited. Models that focus on describing the change of system as a function of some IVs, especially time, are called static models (Hertzog & Nesselroade, 2003; Voelkle, in press). However, it is worth noting that GPPMs can also represent dynamic models. As I have previously shown, there is an equivalent GPPM for any multiple-subject SSM.

In general, the amount of mathematical machinery needed for multiple-subject SSM seems higher than for GPPM. The only advanced concept needed for GPPM are GPs and their descriptions, mean and covariance functions. Mean and covariance functions, however, are simple generalization of mean vectors and covariance matrices. In contrast, the concept of SDEs is rather complicated. It extends the already complex topic of regular differential equations. Thus, GPPM may provide easier access to continuous-time modeling than multiple-subject SSM.

Similarly, as pertains to longitudinal SEM, the fact that multiple-subject SSM and GPPM originate from different communities and have consequently developed different default models can be leveraged. Furthermore, as I have noted earlier, there are models that can be expressed with GPPM but not with multiple-subject SSM. In the next section, I will provide an example for the way the exponential squared covariance function, which is popular within machine learning and cannot be expressed using multiple-subject SSM, is a viable alternative to the AR(1) model, a multiple-subject SSM which is popular within psychology (Hertzog & Nesselroade, 2003).

To summarize, both multiple-subject SSM and GPPM are viable methods for continuous-

time modeling. However, GPPM is more general because every multiple-subject SSM can be represented as a GPPM but not vice versa. An important representative of the models that can only be expressed using GPPM is again the exponential squared covariance model. While different algorithms are employed to derive the results, the central forms of inference are conceptually identical in GPPM and multiple-subject SSM. An important difference between GPPM and multiple-subject SSM is the language of model specification. In GPPM, a GP is defined via its mean and covariance functions, whereas it is described via a SDE in multiple-subject SSM. Thus, multiple-subject SSM allows dynamic description a model, i.e., by making assumptions about the form of the change from one time point to another, whereas the model is described in static fashion in GPPM, that is, as a function of the IVs.

4.2. Demonstration of Gaussian Process Panel Modeling

This section serves two purposes. First, I demonstrate how GPPM and more specifically the GPPM toolbox can be used for data analysis. Second, I present concrete examples for some of the advantages of GPPM over conventional panel modeling techniques introduced in the previous section.

This section is divided into two parts. The first is devoted to exploring the usefulness of the exponential squared covariance function for psychological data analysis. As I have already mentioned, this covariance function encodes the assumption that the process under investigation is smooth. It is popular within the GPR community. I will show that the exponential squared GPPM is similar to the continuous-time AR(1) model, which is commonly used in psychology. Importantly, however, the AR(1) model leads to continuous but not smooth processes. To explore whether the exponential squared model is a viable alternative to the AR(1) model, I will compare the GPPM expressed by the exponential squared covariance function against the continuous-time AR(1) model based on a longitudinal panel study (Heitmeyer, 2004), in which authoritarianism was measured repeatedly. Data from this study have previously been analyzed with a continuous-time AR(1) model using multiple-subject SSM (Voelkle et al., 2012). I will demonstrate that the exponential squared GPPM is selected over the continuous-time AR(1) model on the authoritarianism data set.

In the second part, I show how GPPM can be used to easily avoid the often unrealistic assumption of uncorrelated errors when using the LGCM. To this end, I implement the LGCM with AR(1)-correlated errors using GPPM. While this model can also be expressed using both extended SEM and multiple-subject SSM, I argue that GPPM provides the easiest approach. The reason for this is that it is well suited for static (Hertzog & Nesselroade, 2003; Voelkle, in press) continuous-time modeling. I will also demonstrate that the LGCM with AR(1)-correlated errors is selected over the regular LGCM on a data set originating from the COGITO study (Schmiedek, Bauer, Lövdén, Brose, & Lindenberger, 2010). COGITO participants underwent an extensive cognitive training, during which their positive affect was also assessed. I concentrate on the positive affect data here.

4.2.1. Exponential Squared Covariance Function as Alternative to the Autogressive Model

Foundations

Again, in this section I will explore wheter the exponential squared GPPM is a viable alternative to the AR(1) model. I will first introduce the discrete-time AR(1) model, then continue with the continuous-time AR(1) model and its GPPM representation, the exponential covariance function. After that, I will present the exponential squared covariance function in detail and relate it to the exponential covariance function.

The conventional AR(1) model is a discrete-time model. Let $Y_{i,j}$ be the random vector representing the jth time point for the ith person, then the AR(1) model can be

represented by the following formula:

$$Y_{i,j} = c + aY_{i,j-1} + \epsilon_{i,j},$$

where c, a are scalars and $e_{i,j} \sim \mathcal{N}(0, \sigma^2)$. Within psychology, the AR(1) model is typically represented as a SEM.

The discrete-time AR(1) model can be interpreted as a special case of the continuous-time AR(1) model for which all intervals between two successive measurements are the same. The continuous-time AR(1) model is typically represented as a SSM, that is, in the form of the following SDE

$$\frac{\mathrm{d}L_i(t)}{\mathrm{d}t} = c + aL_i(t) + \epsilon_i(t). \tag{4.2.1}$$

c, a are scalars again and $\epsilon_i(t) \sim \mathcal{GP}(0, \delta(s-t)\sigma^2)$ is a white noise GP. Since Equation 4.2.1 is a special case of the state equation, as used in a SSM, it describes a set of GPs. The set of GPs can equivalently be described by parameterized mean and covariance functions, that is, a GPPM

$$L_i(t) \sim \mathcal{GP}(m(t), k(s,t)).$$

Assuming that the value of the process at some arbitrary time point t_0 is y_{t_0} for everyone, the corresponding mean and covariance functions for all time points $t_0 > 0$ are

$$m(t) = -\frac{c}{a} + \left(y_{t_o} + \frac{c}{a}\right) e^{at} \qquad k(s,t) = \frac{\sigma^2}{-2a}\left(e^{a|s-t|} - e^{a(s+t)}\right).$$

a has to be smaller than 0 as otherwise $k(s,t)$ would not be a valid covariance function.

In the stationary variant of the AR(1) model it is assumed that the process has been observed for a long time. That is, the behavior of the process for large time values t is of interest. This is achieved by letting t and s go to infinity:

$$\lim_{t\to\infty} m(t) = -\frac{c}{a} \qquad \lim_{\substack{s\to\infty \\ t\to\infty}} k(s,t) = \frac{\sigma^2}{-2a} e^{a|s-t|}.$$

The mean and covariance functions can be simplified, by replacing $c = -\mu a$, where $\mu \in \mathbb{R}$, and $\sigma^2 = -2\sigma_\epsilon^2 a$, with $\sigma_\epsilon \in \mathbb{R}$:

$$m(t) = \mu \qquad k(s,t) = \sigma_\epsilon^2 e^{a|s-t|}.$$

The covariance function can be further reparameterized to

$$k(s,t) = \sigma^2 \exp\left(-\frac{|s-t|}{\rho}\right),$$

by setting $a = -1/\rho$ with $\rho \in \mathbb{R}^+$. This covariance function is known as the exponential covariance function in the GPR community, and known to represent the continuous-time AR(1) model if the IV is time.

The exponential covariance function is not particularly popular for GPR. In contrast to that, an initially similar covariance function, the exponential squared covariance function, is among the most used covariance functions. The exponential squared covariance function is commonly presented as

$$k(s,t) = \sigma^2 \exp\left(-\frac{(s-t)^2}{2l^2}\right),$$

with $\sigma^2 > 0$ and $l > 0$. It can equivalently be written as

$$k(s,t) = \sigma^2 \exp\left(-\frac{(s-t)^2}{\rho}\right),$$

with $\sigma^2 > 0$ and $\rho > 0$. Thus, the exponential covariance function and the exponential squared covariance function differ in the fact that only for the latter the distance $|s-t|$ is squared.

A different perspective on the differences between the two covariance functions can be obtained by comparing the model-implied auto-correlations. Let Y_s, Y_t be two random variables from a GP that are $r = |s-t|$ time units apart. The variance for each of these variables under both covariance functions is simply σ^2. It is the implied correlation in which the two covariance functions differ. For the exponential covariance function the implied correlation is $\text{Corr}(Y_s, Y_t) = \exp(-\frac{r}{\rho})$, whereas it is $\text{Corr}(Y_s, Y_t) = \exp(-\frac{r^2}{\rho})$ for the exponential squared covariance. The model-implied correlation for the latter can be written as

$$c(r)^r,$$

where $c(r) = \exp(-\frac{r}{\rho})$ is the model-implied correlation of the exponential covariance function. The ratio of two covariance function is thus

$$\frac{c(r)^r}{c(r)} = c(r)^{r-1}.$$

Thus, the correlation implied by the exponential squared covariance function for the same length scale parameter ρ is higher for a time distance of $r < 1$, lower for $r > 1$, and identical to the correlation implied by the exponential covariance function for $r = 1$. The differences grow exponentially the further away from 1 the distance r is.

In Figure 4.2, I provide a graphical illustration of the differences. The sample realizations of the exponential and the exponential squared covariance function provide a good illustration of a smooth and a non-smooth trajectory. While both realizations are continuous (no jumps), the derivative (i.e., the change) of the realization from the exponential covariance function clearly is not continuous, that is, the realization is not smooth, whereas the first and the second derivative (and all other derivatives) are continuous for the realization of the exponential squared covariance function, making it smooth.

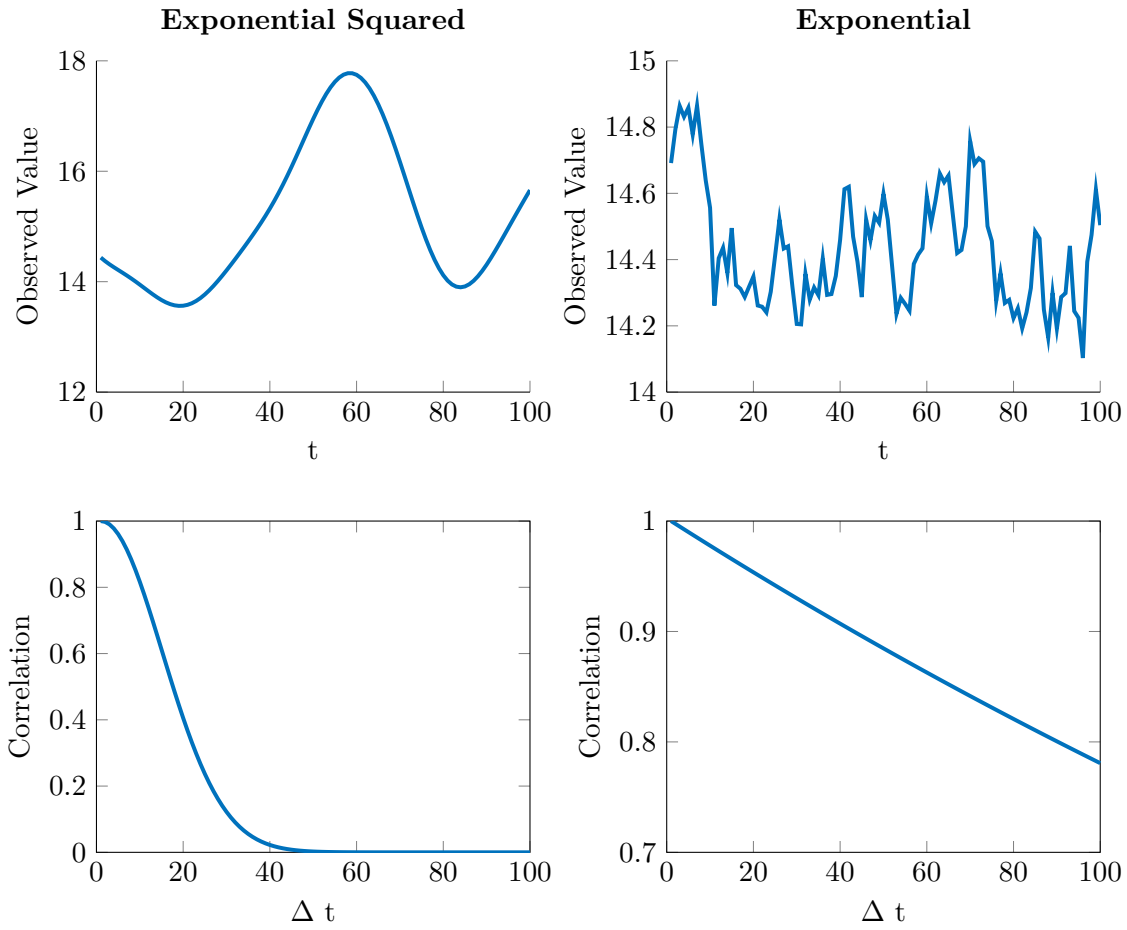


Figure 4.2.: Graphical illustration of the differences between the exponential squared and the exponential covariance function. In the upper two panels, a corresponding realization of each covariance function is shown. The lower two panels display the model-implied correlations for two time points that are Δt apart. To generate the data, I set $\sigma^2 = 2$ and $\rho = 400$.

Demonstration Data Set: Stability of Authoritarianism

I reuse data that were analyzed by Voelkle et al. (2012) to motivate the use of the continuous-time AR(1) model to compare the continuous-time AR(1) model (exponential covariance function) and the exponential squared covariance function. The data originate from a German panel study (Heitmeyer, 2004), measuring people aged 16 years and older who do not have an immigration background using computer-assisted interviews. Measurements were performed in $2002, 2003, 2004, 2006$, and 2008, but not in 2005 and 2007.

Among other variables, authoritarianism was measured. According to Voelkle et al. (2012, p. 24–26),

> ... to date, most researchers ... agree that authoritarianism reflects a) an individual preference for submission under authorities (authoritarian submission), b) a strict orientation along the perceived conventions of the ingroup (authoritarian conventionalism), and c) aggressive stances toward outgroups (authoritarian aggression). ....
>
> Authoritarianism was measured by four items .... Items were presented on a 4-point rating scale providing response options from 1 (agree totally) to 4 (do not agree at all). The original response options were recoded, so that higher values indicate higher agreement. The item wording was "In order to preserve law and order, it is necessary to act harder against outsiders," "One should punish criminal acts harder," "One should be obedient and respectful to authorities," and "One should be grateful to leaders who tell us what to do." The average of the four items was used for all subsequent analyses.

$N = 2,722$ people took part in the study. Response rates were 100% in the first wave, 43% in the second wave, 30% in the third wave, 48% in the fourth wave, and 21% in the last wave. While the dropout rate is substantial, it is typical for longitudinal surveys.

Employed Exponential and Squared Exponential Model

Here, I present the details of the exponential as well as the exponential squared model as used for the analysis of the authoritarianism data. For both models, I wanted to avoid stable between-person differences in the mean influencing the results. Therefore, I started with the following model that allows for person-specific means:

$$m(t) = \mu_a, \qquad k(s,t) = \sigma_a^2.$$

For every person i, a person-specific mean a_i is modeled. They are assumed to be realizations of a Gaussian distribution $\mathcal{N}(\mu_a, \sigma_a^2)$. For the exponential model, I added the exponential covariance function to the covariance function, and for the exponential squared model, I added the exponential squared covariance function. This leads to the following GPPM:

Measure	Exponential Model	Exponential Squared Model	Ratio
AIC	10846.59	**10820.84**	1.002
BIC	10876.14	**10850.39**	1.002
nCV	10846.54	**10821.99**	1.002

Table 4.1.: Akaike information criterion (AIC), Bayesian information criterion (BIC), and negative cross-validated log-likelihood (nCV) for the exponential and the exponential squared model. The ratio is exponential / exponential squared. Bold face marks the model selected on the basis of the corresponding measure. The smaller value of a measure indicates which model to select.

$$L_i(t) \sim \mathcal{GP}(\mu_a, \sigma_a^2 + k_m(s,t)),$$

where $k_m(s,t)$ represents the part of the covariance function specific to the particular model. For the exponential model it was the exponential covariance function

$$k_m(s,t) = \sigma^2 \exp\left(-\frac{|s-t|}{\rho}\right), \tag{4.2.2}$$

and for the exponential squared model the exponential squared covariance function

$$k_m(s,t) = \sigma^2 \exp\left(-\frac{(s-t)^2}{\rho}\right). \tag{4.2.3}$$

I included a term allowing for measurement error for both models. Thus, the GPPM for the observations is a mixture of the latent process $L_i(t)$ and a white noise GPPM such that

$$Y_i(t) = L_i(t) + \epsilon_i(t)$$

with $\epsilon_i(t) \sim \mathcal{GP}(0, \delta(s-t)\sigma_\epsilon^2)$.

Results

To investigate which model is better suited for the authoritarianism data, I compared the exponential and the exponential squared model using three model-scoring methods, the AIC, the BIC, and the negative cross-validated log-likelihood as introduced in Section 3.7.5. For cross-validation, I used 10 folds. The scores for both models and the three scoring methods are shown in Table 4.1. They do not differ substantially between the two models. However, the exponential squared model is favored by all three, providing evidence that it is worth considering as an alternative to the exponential model, which is widely used within psychology.

Did the parameter estimates differ by model? Note that the meaning of the length scale parameter ρ is substantially different for the exponential squared and the exponential covariance functions. Comparing ρ between the two models is similar to comparing the

Parameter	Lower Bound	Estimate	Upper Bound
μ_a	2.82	2.85	2.87
σ_a^2	0.00	0.00	0.11
σ^2	0.37	0.47	0.50
ρ	13.24	13.42	15.26
σ_ϵ^2	0.04	0.05	0.06

(a) Exponential Model

Parameter	Lower Bound	Estimate	Upper Bound
μ_I	2.82	2.85	2.87
σ_a^2	0.21	0.26	0.30
σ^2	0.16	0.19	0.23
ρ	20.95	21.39	30.54
σ_ϵ^2	0.07	0.08	0.08

(b) Exponential Squared Model

Table 4.2.: 95%-confidence intervals (CIs) as well as maximum likelihood (ML) estimates for the parameters from exponential and the exponential squared model.

slope parameter b of a linear model bx with the coefficient c of a quadratic model cx^2. The remaining parameters are comparable between the two models.

The parameter estimates and their corresponding 95%-confidence intervals (CIs) are displayed in Table 4.2. The estimates for the mean parameter μ_a are identical, whereas the estimates for the variance of the mean σ_a^2, the variance σ^2, and the error variance σ_ϵ^2 are all different. Conclusions based on the exponential model would thus be substantially different in comparison to those based on the exponential squared model.

Using GPPMs, one can also compare the person-specific predictions obtained by the different models. This is possible for any combination of GPPMs even if the parameters are not comparable. In Figure 4.3, I show the person-specific predictions using both models for one person. Predictions for both the observable process $Y_i(t)$ and the latent process $L_i(t)$ are shown. While the predictive distributions are relatively similar, there are substantial differences between them. Most notably, the predictive mean and the predictive variance for the exponential squared model are both smooth, as desired by the "nature does not jump" assumption, whereas this is not the case for the exponential model.

To summarize, the exponential squared model, which implements the "nature does not jump" assumption, and is only expressible via GPPM is selected over the AR(1) model on the authoritarianism data set. This is the first empirical evidence that the exponential squared model may be important for psychology.

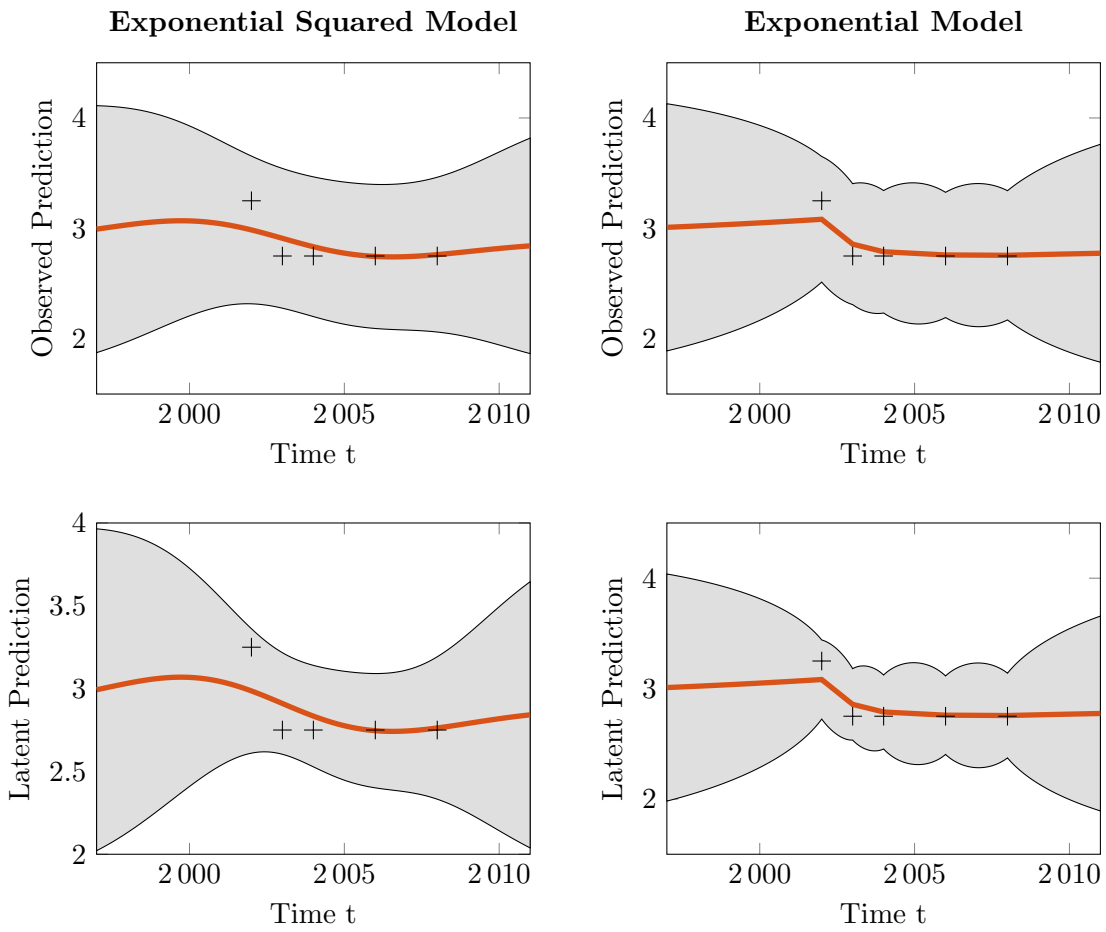


Figure 4.3.: Person-specific predictions of the exponential squared and the exponential model for one person i. The predictions for the observable process $Y_i(t)$ are shown in the first row. The predictions for the latent process $L_i(t)$ are shown in the second row. The bold line indicates the mean of the predictive distribution for every time point. The grey area displays the 95% credibility region.

4.2.2. Extending LGCMs With Autocorrelated Error Structures

In this section, I show how GPPM can be used to easily avoid the often unrealistic assumption of uncorrelated errors when using the LGCM. To this end, I implement the LGCM with AR(1)-correlated errors using GPPM.

Latent Growth Curve Model

The general idea of the LGCM is that within each person the variable of interest develops according to a linear trend. Its slope and intercept may vary between persons according to a Gaussian distribution. Formally, this can be expressed as follows

$$\begin{aligned} &Y_i(t) = a_i + b_i t + \epsilon_i(t) \\ &a_i \sim \mathcal{N}(\mu_a, \sigma_a^2), \quad b_i \sim \mathcal{N}(\mu_b, \sigma_b^2) \quad \epsilon_i(t) \sim \mathcal{N}(0, \sigma_\epsilon^2) \\ &\mathrm{Cov}(a_i, b_i) = \sigma_{ab} \quad \mathrm{Cov}(\epsilon_i(s), e_j(t)) = 0 \text{ if } s \neq t \text{ or } i \neq j \end{aligned} \tag{4.2.4}$$

The parameters of the model are thus $\theta = [\mu_a, \sigma_a^2, \mu_b, \sigma_b^2, \sigma_{ab}, \sigma_\epsilon^2]$. For every parameter value $Y_i(t)$ is a GP. Thus, the model for the GP $Y_i(t)$ can be represented by mean and covariance functions as follows. I have already shown the mean and covariance functions representing a LGCM in Example 3.6.4. Here, I repeat the result for convenience.

$$\begin{aligned} &m(t;\theta) = \mu_a + \mu_b t \\ &k(s,t;\theta) = \sigma_a^2 + \sigma_{ab}(s+t) + s\sigma_b^2 t + \delta(s-t)\sigma_\epsilon^2 \end{aligned}$$

Latent Growth Curve With AR(1) Error

The assumption that the error process is a white noise process, that is, there is no correlation between the error terms at different time points, might not always be reasonable for the situations in which the LGCM is applied. Indeed, correlations between error terms often emerge in panel data (Sivo, Fan, & Witta, 2005). The main problem in modeling panel data with error correlations when applying a model that does not assume such correlations is that it introduces systematic bias in the parameter estimates (Sivo et al., 2005). The solution to this problem is to include correlations among the error terms in the model.

The most popular approach to introducing correlations among the error terms in discrete-time modeling is to use a stationary AR(1) process among the error terms. That is, the covariance between two error terms $\epsilon_{i,j}$ and $\epsilon_{i,j+r}$ that are $r \in \mathbb{N}$ units of time apart is

$$\mathrm{Cov}(\epsilon_{i,j}, \epsilon_{i,j+r}) = \sigma_\epsilon^2 a^r$$

with $|a| < 1$. When using any SEM program this approach can easily be implemented by adding regressions on the error terms.

The continuous-time AR(1) model has to be employed to account for AR(1)-error correlations in continuous-time models. This can be implemented by using an extended SEM software such as OpenMx (Neale et al., 2016). However, setting up the model

becomes relatively complex, as it relies on definition variables and nonlinear transformations of parameters. In contrast, as I will show here, changing the error process from a white noise process to a stationary AR(1) process is straightforward using the GPPM toolbox. Exchanging the default uncorrelated error process for another process, like the exponential squared process, is also easy.

To implement a continuous-time AR(1) error structure using GPPM, the white noise covariance function $\delta(s-t)\sigma_\epsilon^2$ is replaced by the exponential covariance function

$$k_{\exp}(s,t) = \sigma_\epsilon^2 \exp\left(-\frac{|s-t|}{\rho}\right), \quad \rho > 0,$$

such that the new covariance function, representing a LGCM with AR(1)-correlated errors, is

$$k(s,t;\theta) = \sigma_a^2 + \sigma_{ab}(s+t) + s\sigma_b^2 t + \sigma_\epsilon^2 \exp\left(-\frac{|s-t|}{\rho}\right).$$

To implement any other error structure (for example the exponential squared function) the appropriate covariance function has simply to be used instead of the exponential covariance function.

Note that the two limiting covariance functions of the exponential covariance function are the white noise covariance function

$$\lim_{\rho \to 0} k_{\exp}(s,t) = \delta(s-t)\sigma_\epsilon^2$$

and the constant covariance function

$$\lim_{\rho \to \infty} k_{\exp} = \sigma^2. \tag{4.2.5}$$

Thus, the LGCM with uncorrelated errors is nested within the LGCM with an AR(1) error, with the constraint $\rho = 0$. Figure 4.4 visualizes the different error structures, by presenting a sample from one LGCM with the three different error structures: The AR(1) structure (moderate ρ values), the white noise structure ($\rho = 0$) and the constant structure ($\rho = \infty$).

Model Specification

Since one of the main points of this section is that it is easy to define different error structures for a LGCM using GPPM, I will also present how the model is specified in practice using the GPPM toolbox. Remember that model specification for GPPMs consists of specifying parameterized mean and covariance functions.

I start with the specification of the mean and the covariance function representing the LGCM without a measurement error. The mean function

$$m(t;[\mu_a,\mu_b]) = \mu_a + \mu_b t$$

consists of an addition of a constant (μ_a) and a linear mean ($\mu_b t$). In MATLAB code it is specified as follows:

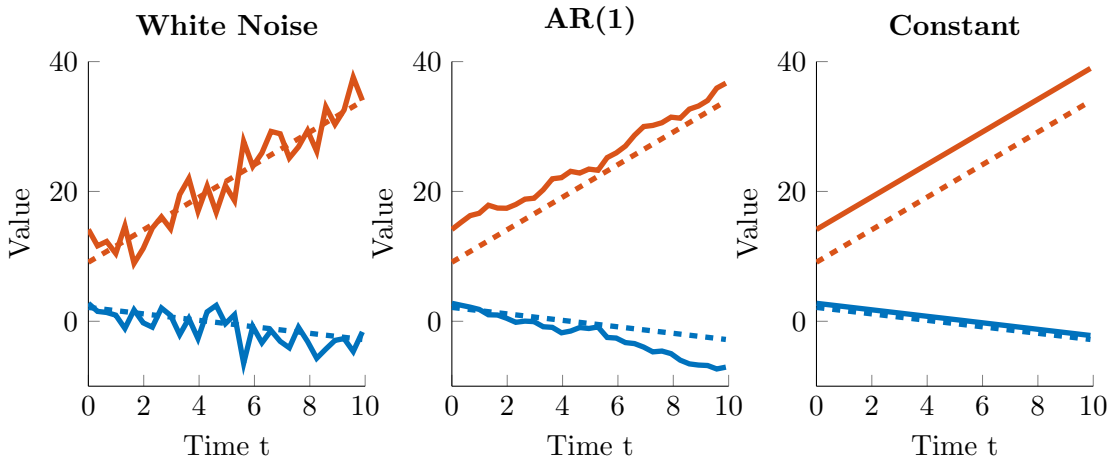


Figure 4.4.: Illustration of the different error structures: the standard white noise error process ($\rho = 0$), the continuous-time AR(1) error structure ($\rho = 10$) and the constant error process ($\rho = \infty$). The same two example trajectories were generated from a LGCM without error for each plot. The resulting trajectories are indicated by dashed lines in each plot and the same for all three plots. The plots differ in the error added to the error-free LGCM. The solid lines denote the resulting trajectories. The plot demonstrates that under the different error structures the observations (solid line) differ, even though the latent concept of interest (dashed line) is identical.

```
LGCMMean ={@meanSum,{@meanConst @meanLinear}};
```

This reads that the mean function is a sum (@meanSum) of a constant mean(@meanConst) and a linear mean (@meanLinear) function.

The LGCM covariance function without the measurement error

$$k(s,t) = \sigma_a^2 + \sigma_{ab}(s+t) + s\sigma_b^2 t$$

can also be expressed as a sum of different covariance functions. σ_a^2 is simply a constant covariance function. $s\sigma_b^2 t$ can be expressed as a scaled version of the linear covariance function (st). For the last term $\sigma_{ab}(s+t)$, I had to implement a new covariance function, which I named covCorr. In MATLAB code the specification of the covariance function is as follows

```
LGCMCov = {@covSum, {@covConst, {@covScale {@covLIN}} , @covCorr}};
```

This reads that the covariance function is a sum (@covSum) of the constant covariance function (@covConst), a scaled version of the linear mean function ({@covScale {@covLIN}}), and the covCorr covariance function, which represents the correlation between the intercept and the slope.

To add the measurement error, the white noise covariance function $k(s,t) = \delta(s-t)\sigma_\epsilon^2$ has to be added.

```
LGCMCovError = {@covSum, {LGCMCov, @covNoise}};
```

This reads that the covariance function is a sum (@covSum) of the covariance function for the latent growth curve model without error (LGCMCov), as previously defined, and the white noise covariance function (@covNoise).

To implement the continuous-time AR(1)-error assumption, the white noise covariance function has to be replaced with the exponential covariance function. The exponential covariance function is a special case of the so-called Matérn covariance function. The MATLAB code for the specification of it is:

```
expCov = {@covMaterniso 1};
```

This reads that the exponential covariance function is the Matérn covariance function with $d = 1$, and respectively $\nu = 1/2$ (for details, see Rasmussen [2006, Section 4.2]). Thus, the code generating the full covariance function for the LGCM with AR(1)-correlated errors is:

```
covF = {@covSum {LGCMCov expCov}};
```

The mean and the covariance function are saved in a model object.

```
model.meanf = LGCMMean;
model.covf = covF;
```

In principle, the definition of the mean and covariance functions is sufficient to specify a GPPM. However, the iterative optimization algorithm used to obtain the ML estimate requires starting values for all parameters. These are provided as follow: Note that the mean function has two parameters (the mean of the intercept μ_a and the mean of the slope μ_b), and the covariance function has five (the variance of the intercept σ_a^2 and the slope σ_b^2, the covariance between the intercept and the slope σ_{ab}, the error variance σ_ϵ^2, and the length scale ρ).

```
model.hyp.mean = [0 1];
model.hyp.cov = [2 3 0 6 1];
```

This reads that the starting values for the parameter vector $[\mu_a, \mu_b, \sigma_a, \sigma_b, \sigma_{ab}, \rho, \sigma_\epsilon^2]$ are $[0, 1, 2, 3, 0, 6, 1]$.

To obtain the ML estimate, a data set $\{(X_i, y_i) : i \in 1, \ldots, N\}$ is required. The jth row of the matrix X_i contains the time information for the jth observation of person i. The entry y_{ij} of the vector y_i contains the observed value for the jth observation. In MATLAB the sequence of matrices $(X_1, \ldots, X_N)$ is stored in a cell array denoted by X, and the corresponding sequence of vectors $(y_1, \ldots, y_N)$ in a cell array denoted by Y. Joining the model and the data, parameter estimation is then performed as follows.

```
model.X = X;
model.Y = Y;
fittedModel = gpPanel(model)
```

The object fittedModel contains all kinds of information, most importantly the ML estimate for all parameters.

Demonstration Data Set: Trajectories of Positive Affect During Cognitive Training

As a demonstration data set, I use data from the COGITO study (Schmiedek et al., 2010). The aim of the study was to investigate the intra-individual variability and plasticity of cognitive abilities of younger and older adults. To this end, participants underwent an extensive training regime of cognitive abilities. During training, they were measured on an average of 101 time points spread across an average of 158 days.

A set of emotional variables was also measured in addition to cognitive abilities. Positive and negative affect was assessed using the Positive and Negative Affect Schedule (Watson, Clark, & Tellegen, 1988). Here, I only use the positive affect data. The Positive and Negative Affect Schedule assesses positive affect using the question, "Indicate to what extent you feel this way right now, that is, at the present moment", coupled with the adjectives: excited, strong, interested, enthusiastic, proud, inspired, determined, attentive, active, and alert. For each adjective, participants had to indicate to what ex-

Index	LGCM	LGCM + AR(1)	Ratio
AIC	1802.56	**1798.55**	1.002
BIC	1818.25	**1816.86**	1.001
nCV	1805.60	**1800.30**	1.003

Table 4.3.: Akaike information criterion (AIC), Bayesian information criterion (BIC), and negative cross-validated log-likelihood (nCV) for the regular LGCM and LGCM with AR(1)-correlated errors. The ratio is regular LGCM / LGCM with AR(1)-correlated errors. Bold face marks the model selected on the basis of the corresponding measure. The smaller value of a measure indicates which model to select.

tent they feel this way on a scale from "not at all" (0) to "extremely" (7). As a general marker for positive affect, I used the mean across all 10 adjectives such that a higher value indicates higher positive affect.

For simplicity, I limited my analysis to the data from the younger adults ($N = 100$; 51.5% women; age: 20–31, $M = 25.6$; daily sessions: 87–109, $M = 101$; days in study: 116–372, $M = 165$). I limited my analysis to the measurements that occurred within 9 days of beginning training because I expected the strongest change in this period due to the adaptation to the study.

The most important characteristic of COGITO for this demonstration is that the time intervals between two measurement occasions varied both between and within participants, making continuous-time modeling a necessity.

Results

To investigate whether the AR(1)-correlated error assumption is more adequate than the regular white noise error assumption, I compared the regular LGCM and the LGCM with AR(1)-correlated errors based on the three model-scoring methods: negative cross-validated log-likelihood, AIC, and BIC. Additionally, I could perform a likelihood-ratio test for model selection, since the regular LGCM is nested within the LGCM with AR(1)-correlated errors. The scores for the two models are displayed in Table 4.3. All model scores favor the LGCM with AR(1)-correlated errors. Also, the likelihood-ratio test approach clearly favors the AR(1)-correlated errors with a p-value of 0.0142.

To investigate whether using the regular LGCM still leads to the same parameter estimates even though the LGCM with AR(1)-correlated errors is favored, I compared the parameter estimates for the two models. They are shown with corresponding 95%-CIs for both models in Table 4.4. Overall, the parameter estimates are relatively similar. The means of the intercept and the slope are essentially the same for both models. The average starting level of positive affect is roughly 3.59. The 95%-CI for the average slope level μ_b is $[-0.08, -0.03]$. Thus, both models predict that on average positive affect decreases slightly over the first 9 days of the study. Using the ML estimate from

Parameter	Lower Bound	Estimate	Upper Bound
μ_a	3.37	3.58	3.79
μ_b	-0.08	-0.05	-0.03
σ_a^2	0.64	0.90	1.26
σ_b^2	0.00	0.00	0.01
σ_{ab}	-0.03	-0.00	0.02
σ_ϵ^2	0.41	0.46	0.52

(a) Regular LGCM

Parameter	Lower Bound	Estimate	Upper Bound
μ_a	3.38	3.59	3.79
μ_b	-0.08	-0.06	-0.03
σ_a^2	0.57	0.83	1.19
σ_b^2	0.00	0.00	0.01
σ_{ab}	-0.02	0.01	0.03
ρ	0.29	0.54	0.75
σ_ϵ^2	0.43	0.50	0.57

(b) LGCM with AR(1)-correlated errors

Table 4.4.: 95%-CIs as well as ML estimates for the parameters of the regular LGCM and the LGCM with AR(1)-correlated errors.

the LGCM with AR(1)-correlated errors, on average positive affect decreases from 3.59 to 3.15 in that period. Both models predict that there is hardly any between persons variance in the slope parameter. The same applies to the little correlation between the intercept and the slope.

There is a difference between the two models in the variance of the intercept σ_a^2 and the error variance σ_ϵ^2. The estimated intercept variance is higher for the regular LGCM. The respective CI is $[0.64, 1.26]$ for the regular LGCM, whereas it is $[0.57, 1.19]$ for the LGCM with AR(1)-correlated errors terms. The error variance, in contrast, is slightly higher for the latter LGCM.

I do not report person-specific predictions here because I have not yet implemented latent predictions for a LGCM with AR(1)-correlated errors. This remains to be done in the future.

To summarize, I have shown how the GPPM can be used to relax the assumption that the error terms in a LGCM are uncorrelated, a condition that is often not met in practice, and consequently leads to parameter bias. I have also demonstrated that the error structure can be easily replaced in GPPM. Here, I used the continuous-time AR(1) error structure. On the COGITO data set, the LGCM with AR(1)-correlated errors is preferred over the regular LGCM with uncorrelated errors. Also, the estimates for the parameters of the LGCM differ between the two models.

4.3. Fitting Speed Comparison of Gaussian Process Panel Modeling and Structural Equation Modeling Software

One of the differences between SEM and GPPM, which I have identified in Section 4.1.1, is that the model-implied moments are defined explicitly in GPPM while being defined implicitly in SEM. In SEM multiple matrix multiplications have to be performed to obtain the model-implied moments. Obtaining the model-implied moments thus should require less time with GPPM. Also, my initial pilot studies have been promising in terms of the possibility that GPPM may speed up ML estimation as compared to SEM software. I will investigate this topic in this section.

Increasing the speed of ML estimation (abbreviated as fitting speed in the remainder) is important because obtaining ML estimates for longitudinal SEMs is typically slow if the number of time points is large, making analyses on those kind of data sets cumbersome, if not practically impossible. Many time points per person are, for example, typical for diary or experience sampling studies (Bolger & Laurenceau, 2013).

The first part of this section is a theoretical running time analysis of the fitting algorithms used in SEM and the GPPM toolbox. In the second part of this section, I compare the empirical running time of one SEM toolbox and the GPPM toolbox for two important longitudinal SEMs, the LGCM and the AR(1) model.

4.3.1. Theoretical Comparison

Introduction to Run-time Analysis

Before proceeding with the running time analysis, I will provide a short introduction to the topic. For a more elaborate treatment, the interested reader is pointed to Cormen, Leiserson, Rivest, and Stein (2009, Chapters 2 and 3).

The running time of an algorithm is defined by the number of steps it performs until terminating. The number of steps required typically depends on the size and additional properties of the input. For a given input size, one usually calculates the expected running time for the worst-case input. The reason for this is primarily that the worst-case running time is an upper bound for any running time. Additionally, for many algorithms, the worst case is in fact a typical case, and the running time of the average case is often within the same order of magnitude as the worst-case running time (Cormen et al., 2009, pp. 27–28).

In algorithmic complexity theory, the worst-case running time is not calculated exactly. Instead, one rather computes the asymptotic behavior of the worst-case running time by only considering the fastest growing terms. This is done because a sufficiently large input renders the impact of the smaller terms negligible.

As the notational formalism, the big O notation is commonly used. In this thesis, I use the big O notation such that if (the running time of) an algorithm is in $O(g(n))$, the asymptotic worst-case running time of the algorithm is tightly bounded by $g(n)$. This is how the big O notation is commonly used in practice. For a technically correct

definition and an explanation why the definition is abused in practice, see Cormen et al. (2009, Chapter 3).

In the remainder of this section, I will often employ the following rules for the big O notation.

Theorem 4.3.1. Let there be two algorithms A_1 and A_2 with corresponding running times $T_1(n) \in O(g(n)$ and $T_2(n) \in O(h(n))$, then:

1. The running time of executing both algorithms $T_b(n) = T_1(n) + T_2(n)$ is in $O(\max(g(n), h(n))$.
2. The running time of executing algorithm A_1 m times is in $O(m * g(n))$.

Overview Maximum Likelihood Algorithm

The ML estimation algorithm finds the maximum of the likelihood function with respect to the parameter value θ, or equivalently, the minimum of the minus-two log-likelihood function

$$-2LL(\theta; \mathbf{y}) = -2 \sum_{i=1}^{N} \log \left((2\pi)^{-\frac{T}{2}} |\Sigma_i(\theta)|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} (y_i - \mu_i(\theta))^\top \Sigma_i(\theta)^{-1} (y_i - \mu_i(\theta)) \right) \right).$$

The input for the ML estimation algorithm is the observed data set $\mathbf{y}$ and the corresponding GPPM or SEM. For GPPM, the matrix containing all IVs X_i is also required for every person i. A SEM as well as a GPPM imply a mean vector $\mu_i(\theta)$ as well as a covariance matrix $\Sigma_i(\theta)$ for every person i and parameter value θ.

For the sake of simplicity, I assume that the same number of measurements have been performed for every person. The size of the data set can, thus, be described by the number of persons N and the total number of variables measured per person T. For a univariate panel data set, the total number of variables measured T is equal to the number of measurements. For multivariate data, the total number of variables T is equal to the product of the number of measurements and the number of variables assessed. For SEMs, the number of both observed and latent variables is important for the analysis. I will denote the number of all (latent and observed) variables as K. The number of observed variables is equal to the number the total number of variables measured per person T.

The minimum of the minus-two log-likelihood function can typically not be computed exactly. Numerical optimization methods are used as a remedy. As I have already mentioned, there are many different numerical optimization methods. Many SEM toolboxes use a quasi-Newton method (for example, OpenMx [Neale et al., 2016], lavaan [Rosseel, 2012], and Mplus [Muthén & Muthén, 1998–2012]). The GPPM toolbox uses a conjugate gradient method. Both methods start at a value for the ML estimate, and then

iteratively refine the estimate. In each iteration, the computation of the gradient of the minus-two log-likehood

$$\frac{\partial -2LL(\theta, \mathbf{y})}{\partial \theta} \tag{4.3.1}$$

is the central (most complex) computation. The gradient of a function $f(\theta)$ is the vector containing all partial derivatives

$$\frac{\partial f(\theta)}{\partial \theta_p} \tag{4.3.2}$$

(see also Definition 3.8.1). The gradient can either be computed exactly or approximated numerically.

Time Complexity of Computing the Model-Implied Moments

As I will show later, the central ingredients for both the numerical and the exact gradient are the model-implied person-level mean $\mu_i(\theta)$ and covariance matrix $\Sigma_i(\theta)$, denoted as model-implied moments in the remainder.

For GPPMs, the mean function $(x; \theta)$, the covariance function $k(x, x'; \theta)$, and the matrix containing all IVs X_i are needed to compute the model-implied moments $\mu_i(\theta)$ and $\Sigma_i(\theta)$. The model-implied moments are computed by evaluating the mean function $m(x; \theta)$ T times and the covariance function $k(x, x')$ $T(T-1)/2$ times. The computational complexity of evaluating the mean and covariance functions mostly depends on the number of IVs used, that is, the dimensionality of the input space $\mathcal{X}$. The size of the input space does not depend on that of the data set used. Therefore, I will treat the computational cost of evaluating the mean function $m(x)$ or the covariance function $k(x, x')$ as constant, i.e., it is in $O(1)$. Thus, the computational cost of computing the model-implied mean vector $\mu_i(\theta)$ is in $O(T)$, and that of computing the model-implied covariance matrix $\Sigma_i(\theta)$ is in $O(T^2)$

For SEM, the model-implied mean vector $\mu_i(\theta)$ and covariance matrix $\Sigma_i(\theta)$ are determined by the person-level SEM matrices $A_i(\theta)$ and $S_i(\theta)$, $m_i(\theta)$ and F. Individualization of the SEM matrices is typically achieved via the definition variables approach (see Section 3.2.1). The model-implied moments are determined by the following calculations:

$$\begin{aligned}
\mu_i(\theta) &= F(I - A_i(\theta))^{-1} m_i(\theta) \\
\Sigma_i(\theta) &= F(I - A_i(\theta))^{-1} S_i(\theta)((I - A_i(\theta))^{-1})^\top F^\top.
\end{aligned}$$

The calculation of both the person-level mean $\mu_i(\theta)$ and the person-level covariance matrix $\Sigma_i(\theta)$ requires the matrix $F(I - A_i(\theta))^{-1}$, whose calculating involves inverting a $K \times K$ matrix, requiring $O(K^3)$ steps. The remaining calculations necessary for computing the model-implied moments all require fewer steps. Thus, computing the person-specific matrices requires $O(K^3)$ steps for SEMs and is therefore at least one order of magnitude slower then for GPPMs, since $K \geq T$.

Time Complexity of Computing the Gradient Numerically

I will continue with the analysis of the complexity of computing the gradient given the person-level model-implied moments for all persons.

To compute this gradient numerically, the partial derivatives $\frac{\partial f(\theta)}{\partial \theta_p}$ are approximated numerically. There are different approaches to do this, but they all rely on some samples of the objective function $f(\theta)$ from the line $[\theta - a_1 e_p, \theta + a_2 e_p]$, where a_1, a_2 are arbitrary positive scalars and e_p the pth unit vector of the parameter space. Thus, for every parameter the minus-two log-likelihood has to be evaluated a constant number of times. So, if P denotes the number of parameters in a model, the minus two log-likelihood has to be evaluated an $O(P)$ number of times.

The minus-two log-likelihood $-2LL(\theta; \mathbf{y})$ consists of a sum of the person-level log-likelihoods:

$$LL(\theta; y_i) = \log\left((2\pi)^{-\frac{T}{2}} |\Sigma_i(\theta)|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y_i - \mu_i(\theta))^\top \Sigma_i(\theta)^{-1}(y_i - \mu_i(\theta))\right)\right). \tag{4.3.3}$$

For the person-level log-likelihoods, the model-implied moments $\mu_i(\theta)$ and $\Sigma_i(\theta)$ are required. As I have established earlier, computing them for a GPPM requires $O(T^2)$ operations, and $O(K^3)$ operations for SEMs. Given their computation, calculation of the person-level log-likelihood $LL(\theta; y_i)$ requires $O(T^3)$ operations, since it encompasses inverting the model-implied covariance matrix $\Sigma_i(\theta)$ as well as computing the determinant of the model-implied covariance matrix $\Sigma_i(\theta)$. For GPPMs calculation of the inverse and of the determinant of the model-implied covariance matrix thus dominates the computing of the person-level log-likelihood. Hence, the time complexity is $O(T^3)$. For SEMs, in contrast, the computation of the model-implied covariance matrix dominates the time complexity. Thus, the time complexity is $O(K^3)$. As a result, computing the gradient numerically requires $O(PNT^3)$ steps for SEM and $O(PNK^3)$ steps for GPPM.

Time Complexity of Computing the Gradient Exactly

I continue with the complexity analysis of computing the gradient exactly. The gradient of the minus-two log-likelihood function can be derived as the sum of the gradients of the person-level log-likelihoods $LL(\theta; y_i)$. Using the sum rule of differentiation, the gradient of the person-level log-likelihood function $\frac{\partial LL(\theta; y_i)}{\partial \theta}$ can be further decomposed into

$$\frac{\partial LL(\theta; y_i)}{\partial \theta} = \frac{\partial f_1(\theta)}{\partial \theta} + \frac{\partial f_2}{\partial \theta} + \frac{\partial f_3}{\partial \theta}, \tag{4.3.4}$$

with

$$\begin{aligned} f_1(\theta) &= \log\left((2\pi)^{-\frac{T}{2}}\right) \\ f_2(\theta) &= \log\left(|\Sigma_i(\theta)|^{-\frac{1}{2}}\right) \\ f_3(\theta) &= -\frac{1}{2}(y_i - \mu_i)^\top \Sigma_i(\theta)^{-1}(y_i - \mu_i). \end{aligned}$$

The partial derivatives of the first term $\frac{\partial f_1(\theta)}{\partial \theta}$ are 0 because the first term $f_1(\theta)$ does not depend on the value of the parameter θ. Thus, the computational complexity for the first term is $O(1)$.

The partial derivatives of the second term are

$$\frac{\partial f_2(\theta)}{\partial \theta_p} = \operatorname{tr}\left(\Sigma_i(\theta)^{-1}\frac{\partial \Sigma_i(\theta)}{\partial \theta_p}\right). \tag{4.3.5}$$

Given the inverse $\Sigma_i(\theta)^{-1}$ and the partial derivative $\frac{\partial \Sigma_i(\theta)}{\partial \theta_p}$ of the model-implied covariance matrix, the most complex operation is the multiplication of these two matrices. The multiplication does not need to be carried out fully. Only the diagonal elements of $\Sigma_i^{-1}\frac{\partial \Sigma_i}{\partial \theta_p}$ have to be computed. Thus, the second term in Equation 4.3.5 can be computed in $O(T^2)$ steps. I will examine the computational complexity of calculating the partial derivatives $\frac{\partial \Sigma_i(\theta)}{\partial \theta_p}$ and the inverse $\Sigma_i(\theta)^{-1}$ later, since it is also required for the partial derivatives of the third term $f_3(\theta)$, which is the central one.

The partial derivatives for this third term are as follows. To unclutter the notation, I drop the dependency of the model-implied moments on the parameter.

$$\begin{aligned}
-2\frac{\partial f_3(\theta_p)}{\partial \theta_p} =& \frac{\partial y_i^\top \Sigma_i^{-1} y_i}{\partial \theta_p} - \frac{\partial 2 y_i^\top \Sigma_i^{-1}\mu_i}{\partial \theta_p} + \frac{\partial \mu_i^\top \Sigma_i^{-1}\mu_i}{\partial \theta_p} \\
&= y_i^\top\left(-\Sigma_i^{-1}\right)\frac{\partial \Sigma_i}{\partial \theta_p}\Sigma_i^{-1}y_i + \left(\mu_i^\top - 2y_i^\top\right)\left(\Sigma_i^{-1}\frac{\partial \mu_i}{\partial \theta_p} - \Sigma_i^{-1}\frac{\partial \Sigma_i}{\partial \theta_p}\Sigma_i^{-1}\mu_i\right) \\
&+ \frac{\partial \mu_i^\top}{\partial \theta_p}\Sigma_i^{-1}\mu_i
\end{aligned} \tag{4.3.6}$$

Computing the inverse of the model-implied covariance matrix $\Sigma_i(\theta)^{-1}$ requires $O(T^3)$ steps. This can be done once and does not need to repeated for every parameter θ_p. The partial derivatives of the model-implied moments $\frac{\partial \mu_i(\theta)}{\partial \theta_p}$ and $\frac{\partial \Sigma_i(\theta)}{\partial \theta_p}$ do have to be computed for each parameter θ_p. Calculating these partial derivatives differs between GPPM and SEM. For now, I assume that they have already been computed. By executing the computations in favorable order, the most complex operation in Equation 4.3.6 is a matrix vector multiplication. Thus, the complexity of Equation 4.3.6 is $O(T^2)$.

For SEM, the partial derivatives of the model-implied moments can be calculated as follows. The matrices $B_i := (I - A_i(\theta))^{-1}$ and $E_i := B_i S_i(\theta) B_i^\top$ will prove useful to express them (von Oertzen & Brick, 2014). As von Oertzen and Brick (2014) show:

$$\frac{\partial \mu_i(\theta)}{\partial \theta_p} = FB_i\frac{\partial A_i(\theta)}{\partial \theta_p}B_i m_i + FB_i\frac{\partial m_i(\theta)}{\partial \theta_p}$$

and

$$\frac{\partial \Sigma_i(\theta)}{\partial \theta_p} = \left[FB_i\frac{\partial A_i(\theta)}{\partial \theta_p}E_i F^\top\right]^{\text{sym}} + FB_i\frac{\partial S_i(\theta)}{\partial \theta_p}B_i^\top F^\top,$$

with $A^{\text{sym}} = A + A^\top$. The computations of B_i and E_i are both in $O(K^3)$. B_i and E_i only need to be computed once for each person. The calculation of the partial derivatives $\frac{\partial A_i(\theta)}{\partial \theta_p}$ and $\frac{\partial S_i(\theta)}{\partial \theta_p}$ are both in $O(K^2)$. If executed in the correct order, the computations needed for the partial derivative of the model-implied mean $\frac{\partial \mu_i(\theta)}{\partial \theta_p}$ only consists of matrix vector multiplications and vector additions. The most complex multiplication is a $K \times K$ matrix with a K-dimensional vector ($\in O(K^2)$). If again executed in the correct order, the most complex operation for computing the partial derivative of the model-implied covariance matrix $\frac{\partial \Sigma_i}{\partial \theta_p}$ is the multiplication of a $T \times K$ by a $K \times K$ matrix, which is in $O(TK^2)$.

For GPPM, the partial derivatives of the person-level model-implied moments are calculated differently. Together with the specification of mean and covariance functions, the partial derivatives with respect to all parameters are also required. This supplies the partial derivatives of the person-level model-implied moments. Given these, the partial derivatives of the minus-two log-likelihood function can be computed as for SEM, using Equations 4.3.5, 4.3.6, and 4.3.4. In general, as applies to the evaluation of the mean and covariance functions, the partial derivatives can also be obtained in $O(1)$. Hence, the time complexity for computing the partial derivative of the model-implied mean $\frac{\partial \mu_i(\theta)}{\partial \theta_p}$ is $O(T)$ whereas it is $O(T^2)$ for the model-implied covariance matrix $\frac{\partial \Sigma_i(\theta)}{\partial \theta}$.

Summary

To summarize, the pseudo code for calculating the person-level gradient exactly is as follows.

Algorithm 1 Algorithm for exact computation of the person-level gradient. If the O notation has two functions split by | the first function is for GPPM and the second for SEM

```
1:  procedure GRADIENT
2:      ▷ Precompute shared values
3:      Compute μ_i(θ) and Σ_i(θ) ∈ O(T^2|K^3)
4:      Compute Σ_i^{-1} ∈ O(T^3)
5:      For SEM, compute B_i and E_i ∈ O(K^3)
6:      ▷ Compute all partial derivatives
7:      for every parameter θ_p do
8:          Compute ∂μ_i/∂θ_p ∈ O(T|K^2)
9:          Compute ∂Σ_i/∂θ_p ∈ O(T^2|TK^2)
10:         Compute ∂LL/∂θ_p ∈ O(T^2)
11:     end for
12: end procedure
```

Thus, the computational complexity of calculating the gradient exactly is $O(K^3 + PTK^2)$ for SEM, and $O(T^3 + PT^2)$ for GPPM.

The computational complexity of calculating the gradient numerically and exactly for both SEM and GPPM is summarized in Table 4.5.

Type	GPPM	SEM
Exact	$O(N(T^3 + PT^2))$	$O(N(K^3 + PTK^2))$
Numerical	$O(PNT^3)$	$O(PNK^3)$

Table 4.5.: Computational complexity for calculating the gradient numerically and exactly for both SEM and GPPM. T refers to the total number of variables measured per person, N to the number of persons, P to the number of parameters of the model, and K to the number of all variables in a SEM.

Since the number of persons N appears as a linear term in all terms, it can be removed, which leads to Table 4.6.

Type	GPPM	SEM
Exact	$O(T^3 + PT^2)$	$O(K^3 + PTK^2)$
Numerical	$O(PT^3)$	$O(PK^3)$

Table 4.6.: Computational complexity for calculating the gradient numerically and exactly for both SEM and GPPM per person. T refers to the total number of variables measured per person, P to the number of parameters of the model, and K to the number of all variables in a SEM.

The number of all variables K does not exist for GPPM. This makes a comparison of the computational complexities difficult. However, the number of all variables K can be related to the total number of variables measured per person T. In the majority of SEMs, an upper bound for the number of latent variables can be derived using a linear function with the total number of variables measured per person T as input, $K \leq aT+b$. This assumption further simplifies the table comparing the time complexities to Table 4.7.

Type	GPPM	SEM
Exact	$O(T^3 + PT^2)$	$O(T^3 + PT^3)$
Numerical	$O(PT^3)$	$O(PT^3)$

Table 4.7.: Computational complexity for calculating the gradient numerically and exactly for both SEM and GPPM per person. T refers to the total number of variables measured per person and P to the number of parameters of the model. Here, it is assumed that the number of all variables in a SEM K is linearly related to the total number of variables measured per person T.

The table shows that the computational complexity of computing the gradient numerically is identical for both SEM and GPPM. For the exact computation, the computational complexity differs slightly. However, since the number of parameters is typically relatively small compared to the total number of variables measured per person and only contributes linearly, the computational complexity is essentially the same.

Thus, the time complexity of the ML algorithm does not substantially differ between GPPM and SEM. However, the central computations of calculating the model-implied moments and their partial derivatives can be performed faster using GPPM. Therefore, GPPM software may still be significantly faster in practice, as to be shown in the next section.

4.3.2. Empirical Comparison

In the previous section, I have established that the model-implied moments and their partial derivatives can be computed faster using GPPM. Besides this, there may be technical differences in the implementation of the ML algorithm that lead to substantial speed differences.

Methods

As SEM software I chose OpenMx (Neale et al., 2016). This choice was guided by the following rationale. Among the popular SEM software packages are the R (R Core Team, 2015) packages sem (Fox, Nie, & Byrnes, 2015), lavaan (Rosseel, 2012) and OpenMx (Neale et al., 2016), the first widely used SEM software LISREL (Scientific Software International Inc., 2015), CALIS a module of SAS (SAS Institute Inc., 2015), SEPATH a module of Statistica (Statsoft Inc., 2015), and the standalone softwares Mplus (Muthén & Muthén, 1998–2012) and Ωnyx (von Oertzen, Brandmaier, & Tsang, 2015). I only considered open source software, namely, sem, lavaan, OpenMx and Ωnyx, because a detailed profiling of the different parts of the ML estimation algorithm required modifications in the code. At time of writing, Ωnyx was not suited for running time analysis, as it does not feature a documented command line interface. Only a graphical user interface is provided. In principle, sem and lavaan would be suited. However, they both do not implement the concept of definition variables, making the implementation of a continuous-time LGCM, which I used as one example SEM for the comparison, cumbersome. Thus, I used OpenMx.

During a pilot study I realized that OpenMx was considerably slower than expected. This was caused by an algorithm that is supposed to speed up the inversion of the $(A(\theta) - I)$ matrix. For an AR(1) model, using the naive implementation proved significantly faster, which is why I used it for all my experiments.

As the GPPM implementation, I used the GPPM toolbox developed for this thesis. Since this thesis introduces GPPM, this is the only implementation of GPPM available.

Besides the total running time, I was interested in measuring the time required to compute the model-implied moments and their partial derivatives, since these are the

terms for which the computational complexity differs between GPPM and SEM. However, since OpenMx uses numerical gradients, the partial derivatives are not computed directly. Thus, I could only measure the running time taken to obtain the model-implied moments. I modified both the GPPM toolbox and OpenMx for this purpose.

I wanted to choose the subset of all possible inputs of the ML estimation algorithm to be of maximum relevance for panel studies while remaining computationally manageable. As representatives for the models, I chose the univariate LGCM and the univariate AR(1) model. These models are among the most frequently applied panel models (Hertzog & Nesselroade, 2003). As representatives for the data sets, I employed simulated data using members of the respective models as defined by random parameter values as generating distributions.

The data sets for the LGCM were generated by drawing a parameter from a distribution $p(\theta)$ for every data set. I specified the probability distribution $p(\theta)$ such that all parameters except the covariance of the intercept and the slope σ_{ab} are mutually independent. As the probability distribution for the mean parameters μ_a,μ_b, I used $\mathcal{N}(0, 1)$. For the variance parameters σ_a^2, σ_b^2, and σ_ϵ^2, I used the probability distribution Gamma$(2, 1)$, where Gamma(α, β) denotes the gamma distribution with shape parameter α and scale parameter β (e.g., Wasserman, 2004, pp. 29–30). For the covariance parameter σ_{ab}, I used the uniform distribution between -1 and 1 to select a value for the correlation. Using the variances for the intercept and the slope, I computed the resulting covariance.

The time points on which measurements are obtained I determined by generating a vector τ_i that encodes the time points of measurement for each person i. For a given data set, I drew the jth measurement time point, that is the entry $\tau_{i,j}$ from the following probability distribution

$$\begin{aligned}&\tau_{i,j} = j + w_j + \epsilon_{i,j}\\ &w_j \sim \text{unif}(0, 0.2), \epsilon_{i,j} \sim \text{unif}(-0.1, 0.1).\end{aligned}$$

Here, unif(a, b) denotes the continuous uniform distribution over the interval $[a, b]$. The rationale for using this probability distribution was as follows. j represents the time point that was planned for a given wave of measurements j. The js are equally spaced between the beginning of the study at time point 0 and the end of the study, which coincides with the number of time points $T - 1$. The random variable w_j represents the deviation of this planned time point of the whole measurements wave and encodes the fact that measurement of a wave often occurs later than planned. $\epsilon_{i,j}$ accounts for the fact that more than a few participants can rarely be measured at exactly the same time point. Thus, the measurements of the participants are spread around the planned measurement of the wave.

As the range of the number of participants N and measurements per participant T, I used $N \in \{10, 50, 100\}$ and $T \in \{10, 100, 200, 300, 400, 500\}$. I generated 100 different data sets for every combination.

For the AR(1) model, I only generated data for one participant, as this model is most often used with time series data. The model for the jth observation was

$$Y_j = c + \varphi Y_{j-1} + \epsilon_j,$$

where $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$ is a white noise process with a constant variance σ^2. I ensured that $|\varphi| < 1$. It follows that the model-implied mean and variance are the same for all time points. They are:

$$\mathbb{E}(Y_j) = \frac{c}{1-\varphi} \qquad \mathrm{Var}(Y_j) = \frac{\sigma^2}{1-\varphi^2}.$$

I used these formulas to determine the model-implied mean and variance for the first time point.

As for the LGCM, I generated data by drawing from probability distributions from within the AR(1) model. To randomly select a probability distribution within the AR(1) model, I drew parameter values from a distribution specified over the parameters $p(\theta)$ such that they were mutually independent. For the c parameter, I drew from a standard Gaussian distribution ($\mathcal{N}(0, 1)$); for the φ parameter, from a uniform distribution ranging from 0.1 to 0.9; and for for the σ^2 parameter, from a Gamma$(2, 1)$ distribution.

As the range of the number of measurements I used $T \in \{10, 100, 200, 300, 400, 500\}$ again and generated 100 different data sets per value. In contrast to the LGCM, I did not use the continuous-time AR(1) model, because it is not as commonly used as the discrete-time model.

Results

Before comparing the fitting speeds, I wanted to ascertain that the parameter estimates obtained by OpenMx and the GPPM toolbox are identical. Indeed, for all but 14 of 1800 (LGCM) and 45 of 600 (AR(1)) of the estimated data sets, the estimates were identical up to a tolerance of 0.01 per parameter. I excluded all data sets for which the parameter estimates did not match from the remaining analysis.

Before I discuss the results, I want to present my expectations. This depends on the optimization algorithm used. The GPPM toolbox uses a conjugate gradient optimizer with analytic gradients. Hence, the expected time complexity is $O(Q_1 N(T^3 + PT^2))$, with Q_1 being the number of iterations. The exact computation of the person-level gradient requires $O(T^3 + PT^2)$ steps, which needs to be taken for each person. This in turn has to be computed in every iteration.

In the version employed for this thesis, OpenMx uses the SLSQP optimizer (Kraft, 1994) with numerical gradients, a quasi–Newton method. Thus, the central computation in each iteration is also the gradient. However, in contrast to the GPPM toolbox, OpenMx approximates the gradient numerically. Thus the expected time complexity of the OpenMx algorithm is $O(Q_2 N(K^3 + PTK^2))$, with Q_2 being the number of iterations. The numerical approximation of the gradient requires $O(K^3 + PTK^2)$ steps. This has to be done in each iteration for each person.

Hence, the time complexity of the GPPM toolbox algorithm is $O(Q_1N(T^3 + PT^2))$ whereas it is $O(Q_2N(K^3+PTK^2))$ for the OpenMx algorithm. Assuming that there are no dramatic differences in the number of iterations required Q_1 and Q_2, the difference is only in the time complexity of calculating the person-level gradient, which is $O(T^3+PT^2)$ for GPPM and $O(K^3 + PTK^2)$ for SEM. By using the relationship $K \leq aT + b$ again, which applies to both the LGCM and the AR(1) model, the time complexity for SEM can also be expressed as a function of the number of time points T and the number of parameters P only. Furthermore, the number of parameters is a constant for both the LGCM and the AR(1) model. Thus, the time complexity changes to $O(T^3)$ for both, which means that at least in the asymptote no differences between the two methods are to be expected.

The averages of the empirical running time of the ML estimation algorithm, as well as the respective 95%-CIs for both methods, are shown in Figure 4.5. For the LGCM, I only show the running time for the number of persons $N = 500$. The results for the other participant numbers are qualitatively the same. For both, the LGCM and the AR(1) model, the running time of the GPPM toolbox is smaller from $T = 100$. For the LGCM, at $T = 500$, the running time is 19437 seconds (or 5.4 hours) in OpenMx compared to 13921 seconds (or 3.9 hours) with the GPPM toolbox. The difference is greater for the AR(1) model. For $T = 500$ the average running time is 160 seconds (OpenMx) compared to 18 seconds (GPPM toolbox).

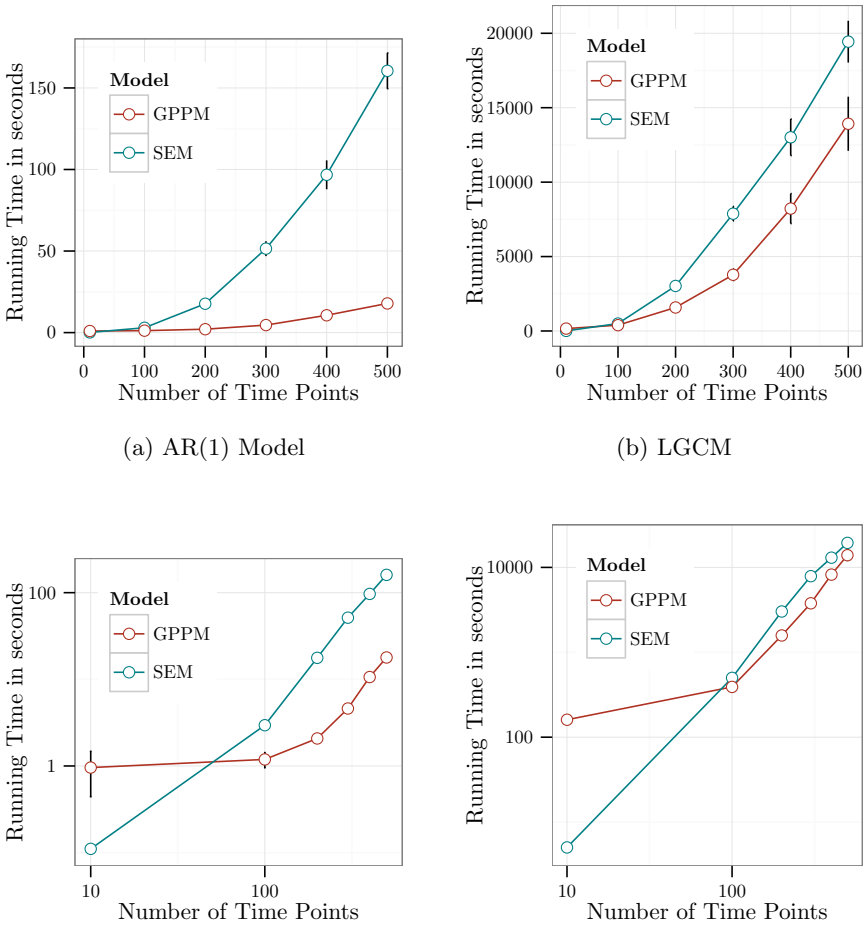
Contrary to my expectations, the running time plots suggest that the asymptotical running time differs between GPPM and SEM for the AR(1) model. However, after changing the scale to a log-log scale, the two lines seem to be parallel at the asymptote, suggesting that the asymptotic running time is the same after all. For the LGCM the same is true. The effect is clearer here, with the lines very close to each other. However, it is worth noting again that in practice, OpenMx is roughly 9 times slower for the AR(1) model and 1.4 times slower for the LGCM for $N = 500$.

As a last point, I wanted to validate the part of the theoretical analysis that revealed that there are differences in the time complexity of calculating the model-implied moments. To do this, I identified the respective code segments in both programs and obtained the time spent in these segments. The theoretical analysis predicts that the time complexity for GPPM is $O(T^2)$ whereas it should be $O(T^3)$ for OpenMx.

The average time spent calculating the model-implied moment for one person is shown in Figure 4.6. For OpenMx, the slopes of the log-log curves are 2.6 for the AR(1) model and 2.5 for the LGCM. This is roughly in the range of 3, which was expected on the basis of the theoretical analysis. For the GPPM toolbox, the slopes of the log-log curves are almost exactly as expected (2), namely 1.9 for the AR(1) model and 2 for the LGCM.

To summarize, using GPPM instead of SEM in order to speed up ML estimation for the AR(1) model seems worthwhile because the expected acceleration is substantial (roughly 9 times faster). For the LGCM, the speedup is not as substantial.

The greatest promise in terms of speeding up ML estimation lies in the many approximation algorithms available for GPR models (e.g., Csató, 2002; Freytag, Rodner, Bodesheim, & Denzler, 2012; Hartikainen & Särkkä, 2010; Särkkä & Hartikainen, 2012)



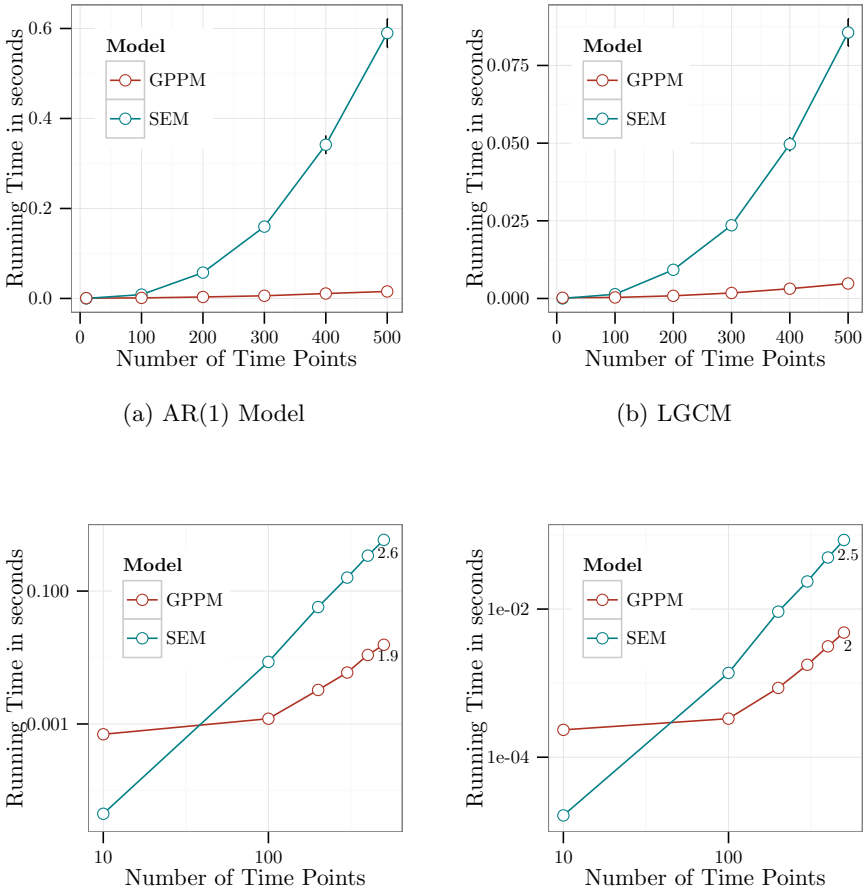
(a) AR(1) Model

(b) LGCM

(c) AR(1) Model Log–log

(d) LGCM Log–log

Figure 4.5.: Empirical running time of the ML estimation algorithm for the two models investigated using the SEM software OpenMx and the GPPM toolbox. The solid lines denote the mean over the up to 100 data sets. The vertical lines denote the 95%-CIs. In the first row, the running time is displayed in a regular, linear scale. In the lower row, the same data are presented but in a log-log scale.



(a) AR(1) Model

(b) LGCM

(c) AR(1) Model Log–log

(d) LGCM Log–log

Figure 4.6.: Empirical running time for computation of the person-level moments for one person for the two models investigated using the SEM software OpenMx and the GPPM toolbox. The solid lines denote the mean over all calculations of the person-level moments. The vertical lines denote the corresponding 95%-CIs. In the first row, the running time is displayed in a regular, linear scale. In the lower row, the same data are presented in a log-log scale.

designed to substantially reduce the running-time at the cost of introducing small imprecision. Investigating the usefulness of theses algorithms for GPPMs will be subject of future work (see also the discussion in Chapter 6).

5. Person-Specific EEG Modeling Based on Supervised Learning

In the previous chapters, I have introduced the novel longitudinal panel modeling approach GPPM, which is based on the supervised learning method GPR. In this chapter, I present a novel EEG analysis method able to estimate person-specific models, and thus, accounts for within-group inter-individual variation in brain–behavior mappings. The method is also based on supervised learning methods, but its inspiration is specifically drawn from analysis approaches as employed in BCI research.

I will briefly recapitulate the rationale for the novel EEG analysis method previously presented in the introduction. The conventional approach, based on estimating ERPs, concentrates on between-group variation. It treats both the intra-individual variation and the inter-individual variation within a group as measurement error. Thus, for the estimated group ERPs to be representative of the ERP for a given person, both the within-group inter-individual variation and intra-individual variation need to be sufficiently small (e.g., Astle & Scerif, 2011; Nagel et al., 2009; Werkle-Bergner et al., 2012). However, empirical data suggests that, at least for children and older adults, the inter-individual variation is not sufficiently small to allow such simplified analysis approaches as they still prevail in the literature.

As one approach to account for this problem, I suggest deriving person-specific models (Molenaar, 2013) that describe the brain–behavior relationship on the person level. To this end, rather than using conventional analysis methods, I employ machine learning methods to increase the signal-to-noise ratio. This is crucial for obtaining person-specific models, as the signal-to-noise ratio cannot be increased by the conventional strategy of averaging across persons. Specifically, I propose using a model selection framework based on nested cross-validation that selects the best brain–behavior mapping out of a set of candidate models and at the same time validates it.

In Section 5.1, I will introduce the supervised learning approach to derive person-specific models. First, I will present the general model selection and validation framework, and then the candidate models employed for the analysis of the WM data set that was used to validate the proposed method. Next, I show how the derived person-specific models can be interpreted spatially on the person level.

In the following section (Section 5.2), I will provide details about the WM data set and about the preprocessing performed prior to applying the method for deriving person-specific models.

I close this chapter with a presentation of the results (Section 5.3). I first show that my proposed method indeed results in person-specific models with better discrimination performance as compared to conventional person-nonspecific models on the WM data

set. I also indicate how the obtained models can be interpreted on a person level as well as on a group level.

5.1. Identifying Person-Specific Models: The Supervised Learning Approach

5.1.1. Foundations

The core idea of my framework is the derivation of person-specific models that optimally discriminate between behavioral conditions and, thus, allow evaluation of the neural underpinnings of interindividual differences in behavioral responses.

In the previous parts of this text, the term model referred to a statistical model. Here, a person-specific model conceptually refers to any mathematical person-specific model relating EEG data to behavior. My framework finds a function $f(x)$ that predicts a specific behavior of a participant based on their EEG data x of a trial. This is an example of a mathematical model and will therefore also be called a model. To obtain the prediction function $f(x)$, learning algorithms are employed. One can describe all of the learning algorithms used here as a combination of a set of candidate models, i.e., a set of prediction functions, in combination with a loss function, which is minimized to select a particular prediction function. Thus, the candidate models used are similar to a statistical model, but a set of predictions functions is described instead of set of probability distributions. In the literature and here, these sets are consequently also often called models. To distinguish the mathematical model $f(x)$ from a learning algorithm used to derive the mathematical model $f(x)$, I refer to the former as an *estimated model* and the latter as a *candidate model* whenever the distinction is not clear from the context.

Person-specific models are estimated models selected from a set of candidate models that vary across multiple dimensions of the observed data space. In EEG, this space typically entails electrode channels, time points, and/or frequencies; but my considerations generally apply to any spatio-temporal method of brain imaging. Candidate models can be derived from a template model class and vary parametrically according to multiple dimensions, first and foremost, to the spatio-temporal segments of the original data they are exposed to. In particular, models operate on different time windows and on subsets of channels or their geometric projections. In the remainder of this subsection, I will describe the proposed framework to estimate person-specific models.

In the following, the number of measured variables per trial will be denoted by M and the number of trials per individual will be denoted by T, as those typically refer to trials ordered in time. For each person, a data set $(x_t, y_t) \in D$ with $t \in \{1, \ldots, T\}$ is measured, which is a set of tuples of observed brain responses $x_t \in \mathbb{R}^M$ and a corresponding dichotomous target variable $y_t \in \{0, 1\}$ that typically corresponds to a given external condition, task, or state. A candidate model, mapping brain responses to the target variables, can then be conceived as a θ-parameterized function $f_\theta(x) = y$, linking the observed neural responses x_t and behavioral states y_t. The specific parameters θ can be estimated by minimizing a loss function on data (usually called the training set).

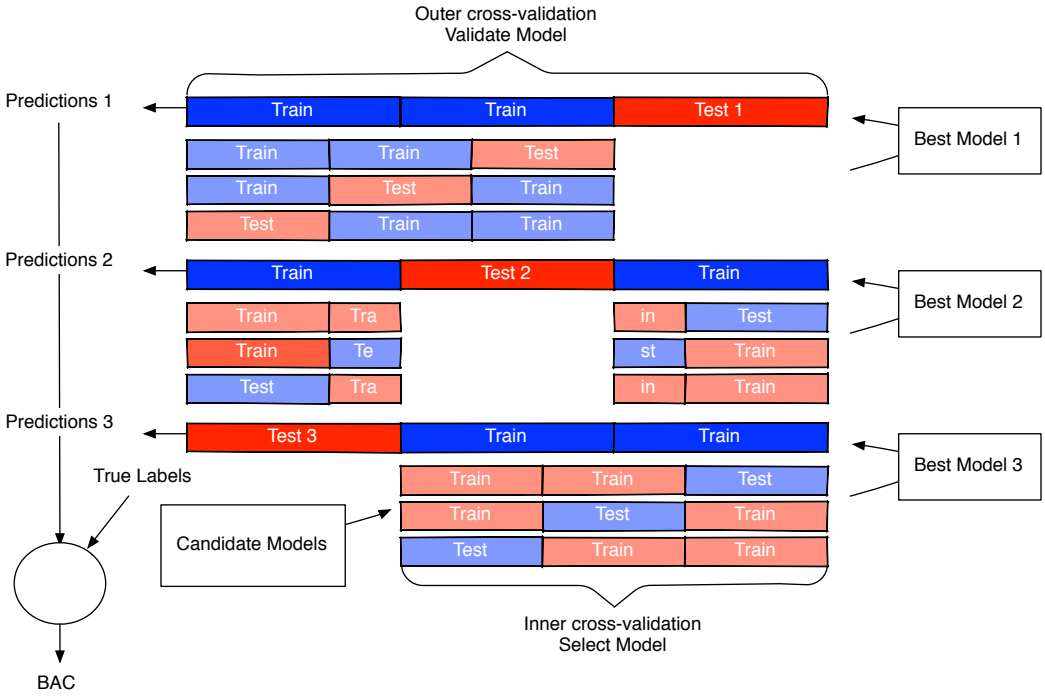


Figure 5.1.: Schematic representation of the nested cross-validation procedure for model selection and evaluation. The schema shows a three-fold nested cross-validation procedure instead of the 10-fold procedure I employed. The model selection takes place in the inner loop (light blue and light red), while model evaluation of the best inner model is performed in the outer loop (dark blue and dark red). The evaluation metric is the balanced accuracy (BAC).

Each estimated model can then be evaluated with respect to its accuracy in predicting a behavioral condition from brain responses, whereby selection of the best model is carried out for each person separately. I propose to use the balanced accuracy (BAC), a loss function accounting for unbalanced target variables that are often encountered in EEG data sets, as the performance measure for each candidate model. The BAC is the average of the accuracies obtained for each target variable state (condition) (Brodersen, Ong, Stephan, & Buhmann, 2010). This metric allows me to select the best of all competing models and interpret the idiosyncratic brain–space information of that model as person-specific information.

To avoid an overoptimistic bias by confounding parameter estimation and model evaluation (Kriegeskorte, Simmons, Bellgowan, & Baker, 2009; Stone, 1974), I estimate the BAC for a given candidate model using 10-fold stratified cross-validation (Kohavi, 1995). The resulting person-specific models can be interpreted as both a measure of interindividual heterogeneity in the sample and as a parsimonious indicator of the location and magnitude of these interindividual differences in brain space. In the remainder of this subsection I will describe the procedure of choosing the best person-specific model in more detail (see also Figure 5.1).

Selection of an optimal model for each person from a set of candidate models entails a cherry picking problem. This cherry picking causes an overoptimistic accuracy estimate if the best accuracy of the model selection phase is reused as the estimate for the overall performance of the selected model (Stone, 1974; Varma & Simon, 2006). To obtain an unbiased BAC estimate, I separately use 10-fold stratified cross-validation for model selection and model validation, which leads to a nested cross-validation procedure with an inner and an outer loop. Nested cross-validation is the de-facto standard procedure for performance evaluation in BCI research (Lemm, Blankertz, Dickhaus, & Müller, 2011). The cross-validated model *selection* takes place in the inner loop, while cross-validated model *validation* is performed in the outer loop. To this end, I randomly partition a data set into ten exhaustive and mutually exclusive subsets. For each of the ten outer cross-validation runs, a temporary training set is formed by leaving out one of the ten folds whereas a temporary test set is formed by the remaining folds. The temporary outer training set forms the basis for the inner cross-validation runs aiming to determine the temporary best model (see below). The performance of the latter is evaluated on the current outer test set. To identify the best model in each run of the outer loop, an inner cross-validation procedure is used. Here again, the available data (i.e., the current outer training set) are split into ten exhaustive and mutually exclusive subsets (i.e, the inner training and test sets for model selection). In each run of the inner loop, the parameters for the candidate models are estimated on the temporary inner training set and their performance is evaluated on the current inner test set. The overall performance estimate for a given candidate model is obtained by aggregating the test set performance across all 10 iterations of the inner loop. The candidate model with the highest average BAC is selected as the current best model and its performance is evaluated on the remaining outer test set.

Technically, the outer cross-validation does not estimate the BAC of the best candidate

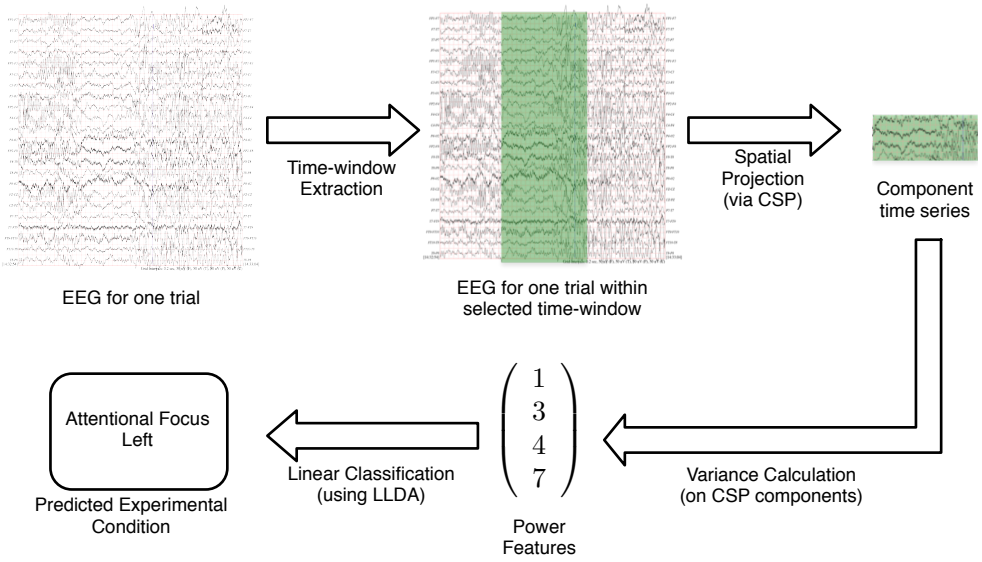


Figure 5.2.: Graphical description of the decision function learnt by the proposed candidate models. Prior to applying this decision function, the EEG is preprocessed. First, the EEG trial is reduced by only considering the signals within a selected time window. After that, the $2k$ selected components of the projection matrix C, as learnt by the common spatial pattern (CSP) method, are applied to the remaining signals. This yields $2k$ component time series. Next, the variance of each of these time series is calculated. The resulting power features are classified by the linear decision function as learnt by Ledoit's linear discriminant analysis (LLDA).

model but rather the BAC of the model selection procedure as a whole. Particularly, in each fold of the outer cross-validation, a different model may be selected as the best one. In other words, nested cross-validation provides an unbiased estimate of the expected BAC when applying the model selection of the inner cross-validation to the whole data set of one person. Put differently, I estimate the expected accuracy of picking a model from my candidate set based on cross-validated predicted accuracy. Consequently, after the nested cross-validation evaluation, the model selection phase (i.e., only the logic of the inner loop) is applied to the complete data set to obtain the best person-specific model. As a final step, the parameters θ of the best model are estimated using the complete data set. The BAC estimate obtained by nested cross-validation is a conservative estimate for the performance of the model using parameters derived from the whole data set.

5.1.2. Candidate Models

To obtain accurate and interpretable brain–behavior mappings, the candidate models that are used in the proposed framework to obtain person-specific models have to be carefully chosen for each data set. The WM data I re-analyzed to test the applicability of

my approach originate from a lifespan study that targeted oscillatory brain mechanisms for WM selection and maintenance in a sample including children, younger, and older adults (Sander et al., 2012). The study used a color change–detection task (Vogel & Machizawa, 2004), in which participants were cued to attend to either the left or the right hemifield and asked to remember the colors of varying numbers of items. Hence, by design, it is possible to identify modulations of rhythmic neural responses that (a) relate to the *attentional focus* and (b) reflect the varying levels of *WM load*. I operationalized (a) attentional focus as the hemifield to which spatial attention should be shifted and (b) WM load as the number of items to be remembered in a change–detection task.

In the following, I describe the set of candidate models I constructed for predicting attentional focus and WM load. Both WM load and attentional focus have been found to be related to power modulations in the alpha range (Kelly, Lalor, Reilly, & Foxe, 2006; Sander et al., 2012; Worden, Foxe, Wang, & Simpson, 2000; Sauseng et al., 2009). To capture differences in signal power I chose the CSP method (Müller-Gerking, Pfurtscheller, & Flyvbjerg, 1999; Ramoser, Müller-Gerking, & Pfurtscheller, 2000). CSP finds a transformation matrix C, mapping the EEG channels onto a set of component projections such that the variances of the resulting time series discriminate optimally between conditions (Ramoser et al., 2000). In order to find the respective transformation matrix C, the eigenvalue decomposition of the mean between-channel covariance matrix is computed. Thus, CSP requires an invertible mean between-channel covariance matrix. If the goal is to discriminate conditions 0 and 1, the projected CSP components are ordered such that the variance of the first component is maximal for condition 0 while being minimal for 1. Vice versa, the last projected component has maximal variance for condition 1 while it is minimal for 0. Hence, the respective components from both ends form complementary pairs with regard to condition prediction. Like other dimensionality reduction approaches such as principal component analysis (PCA), a subset of components can be selected. Due to their complementarity, CSP components are typically picked in pairs, with each subsequent pair adding less predictive information to the overall task. In the following, I refer to the number of these CSP filter pairs as k. To reiterate, a classifier relying on a single filter pair ($k = 1$) bases its prediction on two projected components, for each of which the variance is most informative about the respective target outcome. Consequently, the variances of the EEG projected onto the CSP filters are used as features, that is, as classifier input.

Classification is done by LDA. LDA is a linear classifier, meaning that, its decision function is of the form $\text{sign}(w^\top x + c)$, where x are the features, w is called weight vector, and c bias. For the training of LDA, the feature means and covariance matrices for each class have to be estimated. If the number of trials is large in comparison to the number of features, the sample covariance matrix is a sufficiently precise estimate of the population covariance matrix. For the typical EEG classification problem, the number of trials is roughly equal to or smaller, than the number of features. Hence, the sample covariance matrix is systematically biased (Friedman, 1989). To correct for this bias, regularization is commonly used (Friedman, 1989). The regularization hyper-parameter was set by the analytical solution of Ledoit and Wolf (2004). I refer to the resulting

classifier as Ledoit's linear discriminant analysis (LLDA). The combination of CSP and LDA is commonly used to realize BCIs based on rhythmic neural activity (e.g., Blankertz et al., 2010; Fazli et al., 2009).

A further interest of ours (Karch et al., 2015) was the identification of the most discriminative time window per person. To achieve this, only data from within a single time window are considered for each candidate model. Hence, the models differ with regard to the onset and the duration of the employed time window. The different candidate models were derived from the following settings: Duration of the time windows: $\{100, 150, 200, \ldots, 700\}$ ms, onset of the time window: $\{0, 33, 66, 99, \ldots, L\}$ ms relative to the onset of the memory array (see Section 5.3 for a detailed description of the trial design). The limit L for the onset depended on the duration of the time window. It was chosen such that the latest time windows did not contain signals later than 1000 ms after the onset of the memory array. This was done in order to prevent inclusion of EEG activity evoked by the onset of the test stimulus. As candidates for the number k of the CSP filter pairs I employed $\{2, 3, 4, 5, 6\}$. These candidates were motivated by the rationale of exploring the values around the recommendation to use $k = 3$ by Blankertz, Tomioka, Lemm, Kawanabe, and Mülller (2008). The full grid of all possible combinations of settings was explored, leading to 5 (filter pairs) $\times$ 244 (time windows) = 1220 different candidate models.

5.1.3. Spatial Interpretation of the Best Estimated Model

Based on the features of the candidate models, the best estimated model has the properties: selected time window, CSP matrix C, and LLDA vector w'. Additionally, I used PCA for preprocessing (see Section 5.2.2). Strictly speaking, the PCA matrix P is also part of the model. Interpreting the selected time window is straightforward. The matrices describe how the information contained within the selected time windows was aggregated across the EEG channel, i.e., spatially. Here, I will show how this spatial information can be interpreted as spatial filters and patterns (Bießmann et al., 2012; Parra, Spence, Gerson, & Sajda, 2005).

Filters and patterns assume that the observed data obey a linear measurement model. That is, we assume a set of sources mapped to the observed values by a linear transformation. For raw EEG potentials it is generally assumed that a linear measurement model holds true. A pattern describes the contribution of one source to all electrodes (forward model). A filter describes the linear reconstruction of one source given the observed data (backward model). For a linear classifier the filter simply corresponds to the weight vector w.

To elaborate, the EEG surface potentials $x(t) \in \mathbb{R}^U$ are believed to be a linear mixture of a set of sources $s(t) \in \mathbb{R}^V$ plus noise (e.g. Blankertz, Lemm, Treder, Haufe, and Müller (2011), Bießmann et al. (2012), Parra et al. (2005))

$$x(t) = As(t) + n(t).$$

U denotes the number of channels and V the number of sources. The matrix $A \in \mathbb{R}^{U \times V}$ is called the forward model. Every given column $(a_1, a_2, \ldots, a_V) = A$ describes how

each of the sources contributes to the surface potentials. Therefore, the column a_i of A is called the pattern of source $s_i(t)$. In the following, I will assume that for a given measurement, the time courses of all sources are stored in $S = [s(t_1), s(t_2), \dots S(t_T)]$ and that the time series for all electrodes are stored in $X = [x(t_1), x(t_2), \dots, x(t_T)]$.

If the sources and the EEG surface potentials are known, a task of interest is to find the corresponding backward model. That is, a matrix $W \in \mathbb{R}^{V \times U}$ that recovers the original sources from the observed surface potentials $s(t) = Wx(t)$. An estimate $\hat{W}$ can be obtained by minimizing a distance measure between the original sources $s(t)$ and the reconstructed sources $Wx(t)$. The simplest distance measure is the Euclidean distance. The resulting estimate is called least squares estimate

$$\hat{W}^\top = \arg\min_W \sum_{t=1}^{T} (s(t) - Wx(t))^2 = SX^\top (XX^\top)^{-1}.$$

The rows $(w_1, w_2, \dots, w_s)^\top = W$ of W are referred to as filters, as they describe the contribution of each electrode to a given source.

Typically the sources are not directly observable. Rather, a backward model $\hat{W}$ that optimizes certain properties of the resulting sources is estimated. Independent component analysis (ICA) (Bishop, 2006), for example, is a method to estimate a backward model that optimizes the statistical independence of the recovered sources.

After a backward model $\hat{W}$ has been obtained, the least squares estimate of the corresponding forward model $\hat{A}$ can be derived in analogy to the least squares estimator of the backward model and is hence,

$$\hat{A} = X\hat{S}^\top (\hat{S}\hat{S}^\top)^{-1} = XX^\top \hat{W} (\hat{W} XX^\top \hat{W}^\top)^{-1}, \tag{5.1.1}$$

with $\hat{S} = \hat{W}X$. This estimator is not invariant to constant shifts of the signals or the sources. Constant shifts in a signal originate from a constant source, which generally is not of interest in neuroscience (Parra et al., 2005). Therefore, their mean is subtracted from all rows of X and $\hat{S}$ before the application of Equation 5.1.1. This is equivalent to including a constant source in S and ignoring its parameter estimate.

A particularly easy form of a backward model is a filter $w^\top$, that is, a backward model that only reconstructs one source. Linear classifiers, i.e., parametric classifiers with a linear decision function of the form $f(x) = \text{sign}(w^\top x + c)$, can be regarded as a method to obtain a filter. The optimized property of the reconstructed source is discriminability between two conditions. The filter is simply $w^\top$ and the reconstructed source $w^\top x + c$. The corresponding pattern can be obtained by applying Equation 5.1.1.

Neuroscientific interpretations are typically facilitated by the use of patterns instead of filters (Bießmann et al., 2012; Parra et al., 2005), as patterns are not disturbed by correlated noise sources. Filters and patterns are identical (up to scaling) if $XX^\top$ is a multiple of the identity matrix (see Equation 5.1.1). This is equivalent to the EEG channels being uncorrelated and therefore almost never the case.

In contrast to previous work that employed classifiers to obtain person-specific filter and patterns (Parra et al., 2005; Philiastides & Sajda, 2006), I did not use raw EEG

potentials as features for my classifier. Instead, the model class that I proposed in Section 5.1.2 yields a linear classifier with the variances of the CSP component time series as features. As I will show in the following, this is equivalent to a linear classifier with all entries of the covariance matrix between all channels as features, and thus produces a filter with $\frac{(M+1)M}{2}$ entries, where M is the number of channels.

Theorem 5.1.1. The proposed candidate models, which consist of a combination of PCA, CSP, and LLDA, can be equivalently expressed as a linear classifier with the entries of the within-trial covariance matrix as features.

Proof: Let $X(i) \in \mathbb{R}^{\mathrm{U}\times\mathrm{T}}$ be the EEG within a selected time window observed for trial i and $\hat{\Sigma}(i) \in \mathbb{R}^{\mathrm{U}\times\mathrm{U}}$ the corresponding estimate of the between-channel covariance matrix. Furthermore, let M be the transformation matrix resulting from the composition of PCA and CSP. So, $\mathbb{R}^{2\mathrm{k}\times\mathrm{U}} \ni M := CP$, where C is the transformation matrix learned by the CSP method, P the transformation matrix learned by PCA, and k the number of filter pairs selected for CSP. Let w' be the weights as learned by LLDA. Additionally, let $\mathrm{Var} : \mathbb{R}^{\mathrm{U}\times\mathrm{T}} \to \mathbb{R}^{U}$ be the mapping from a matrix containing U time series of length T to the vector containing the variance of each time series. In addition to that, let $\mathrm{diag} : \mathbb{R}^{n} \to \mathbb{R}^{n\times n}$ be the mapping of a vector to the corresponding diagonal matrix, $\mathrm{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n}$ the mapping of a matrix to the vector containing the entries of its diagonal, $\mathrm{tr}:\mathbb{R}^{n\times n} \to \mathbb{R}$ the trace of a matrix, and $\mathrm{vec} : \mathbb{R}^{n\times p} \to \mathbb{R}^{np}$ the mapping of a matrix to a vector containing the columns of the matrix stacked on top of each other, then the classification function $f_\theta(X(i))$ for trial i is as follows. For notational clarity I drop the dependence of X and Σ on i.

$$
\begin{aligned}
f_\theta(X) &= \overbrace{\mathrm{sign}(w'^{\top}\,\mathrm{Var}(MX)+c)}^{=:f'_\theta(X)} && (5.1.2)\\
\Rightarrow f'_\theta &= w'^{\top}\,\mathrm{diag}(M\hat{\Sigma}M^{\top})+c && (5.1.3)\\
&= \mathrm{tr}(\mathrm{diag}(w')M\hat{\Sigma}M^{\top})+c && (5.1.4)\\
&= \mathrm{tr}(M^{\top}\,\mathrm{diag}(w')M\hat{\Sigma})+c && (5.1.5)\\
&= \underbrace{\mathrm{vec}((M^{\top}\,\mathrm{diag}(w')M)^{\top})^{\top}}_{w^{\top}:=}\underbrace{\mathrm{vec}(\hat{\Sigma})}_{x:=}+c. && (5.1.6)
\end{aligned}
$$

This shows that all candidate models lead to a linear classifier with the entries of the between-channel covariance matrix as features. Hence, w is the filter learnt by my candidate models.

□

For 5.1.3, I made use of the linear transformation theorem of a Gaussian random vector (Theorem A.3.8 in Appendix A.3). For 5.1.5, I employed the fact that the trace is invariant under cyclic permutations. For 5.1.6, $tr(XY) = \mathrm{vec}(X^{\top})^{\top}\,\mathrm{vec}(Y)$ applies.

It follows directly from the linear model assumption for EEG data that the observed covariance data does not comply with a linear model.

Theorem 5.1.2. Under the assumption of a linear model for EEG data, the spatial covariance matrix of the sources is not linearly related to the between-channel covariance matrix of the channels.

Proof: Let $\text{Cov} : \mathbb{R}^{\text{U}\times\text{T}} \to \mathbb{R}^{\text{U}\times\text{U}}$ be the mapping from a matrix containing U time series of length T to the corresponding between-time series covariance matrix. Let $X(i) \in \mathbb{R}^{\text{U}\times\text{T}}$ be the EEG observed for trial i, $S(i) \in \mathbb{R}^{U\times T}$ the activity of the sources and $N(i) \in \mathbb{R}^{U\times T}$ the noise. Dropping the dependence on i, the linear model for EEG can then be formulated as $X = AS + N$. To simplify the proof, I assume that there is no noise, i.e., $N = 0$. It follows that the between-channel covariance matrix $\text{Cov}(X)$ can be related to the spatial covariance of the sources $\text{Cov}(S)$ as follows:

$$\begin{aligned} \text{Cov}(X) &= \text{Cov}(AS) \\ &= A\,\text{Cov}(S)A^\top. \end{aligned}$$

This shows that the relationship between the covariances of the sources and the covariances of the electrodes is quadratic and therefore not linear. This also holds true if we drop the assumption that there is no noise. □

The resulting filter still corresponds to the classifier weights w and the pattern can also still be computed. In order to obtain the pattern corresponding to the classification function learnt by the proposed candidate models I employed Equation 5.1.1, with $X = [\text{Var}(X(1)), \ldots, \text{Var}(X(I))]$ and $\hat{S} = [f'_\theta(X(1)), \ldots, f'_\theta(X(I))]$. Prior to that, I subtracted their means from all rows of X and $\hat{S}$ as explained in earlier.

The interpretation of the filter as forward and the pattern as backward model is no longer valid. However, there is still a meaningful interpretation for both filter and pattern. The weight w_p of the filter corresponding to the pth feature expresses that an increase of the pth feature by one increases the classification score $w^\top x + c$ by w_p. Thus, for positive/negative weights, higher values increase/decrease the classification score and hence, the support for the positive/negative class. Positive class refers to the class that is predicted if $\text{sign}(w^\top x + c) \geq 0$. For my prediction tasks "Attention left" and "Low WM load" are the positive classes and "Attention right" and "High WM load" are the negative classes (see Section 5.2.1).

For the pattern the opposite logic applies. If the classification score increases by one, the expected observed value of the pth feature changes by a_p. As pattern and filter are not invariant to the scaling of the features I normalized both. For illustration purposes I show only the weights corresponding to the variances. Note, however, that for classification and for calculation of the pattern I included all terms, omitting the covariances proved considerably worse (see Section 5.3.4).

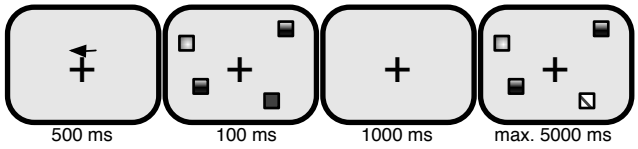


Figure 5.3.: Experimental paradigm. Each trial starts with the presentation of a cue indicating the relevant hemifield. The memory set is presented for 100 ms and followed by a fixed retention interval of 1000 ms. The probe display is shown until a response is given. Different patterns of the squares represent different colors. Adapted with permission from Sander, Werkle-Bergner, and Lindenberger (2012).

5.2. Working Memory Data Set and Preprocessing

5.2.1. Study Design

Here, I will describe the WM data set used to validate my approach. Details on procedures, task design, and EEG data recording can be found in Sander et al. (2012). For convenience, I outline the study design as it is important for the understanding of my analysis approach as well as the results.

Participants

For the present study, I used data from 22 children (M_{age} = 11.9 years, range 10–13 years), 12 younger adults (M_{age} = 24.2 years, range 20–25 years), and 22 older adults (M_{age} = 73.3 years, range 70–75 years). Given that groups of children and older adults are typically more heterogeneous in comparison to younger adults, sample sizes for these two age groups were larger in the initial report. The initial sample included 31 children, 19 younger adults, and 31 older adults. Details about exclusion criteria and descriptive marker tests documenting the age typicality of the sample can be found in Sander et al. (2012). The ethics committee of the Max Planck Institute for Human Development, Berlin, approved the study.

Experimental Paradigm

The exact procedures are described in Sander et al. (2012). In short, a hemifield version of the change–detection task (Vogel & Machizawa, 2004) was used to probe age differences in visual WM capacity (see Figure 5.3). Memory arrays of colored squares were presented to the participants for 100 ms or 500 ms. Targets were defined as the squares presented in the hemifield indicated by a centrally placed cue before each trial. To keep the difficulty of the task comparable for the different age groups, memory arrays of 2, 4, or 5 targets were presented to younger adults while they involved 2, 3, or 4 targets for the older adults and children. Besides the number of targets, the experimental procedure was identical for all age groups. For the present work, I only analyzed the

common set sizes 2 and 4. In the following, they will be referred to as low (2) and high (4) levels of WM load. Trials were presented in four blocks. The first two and the last two blocks were always presented at the same presentation time, and the order of the presentation time was counterbalanced across participants. Set sizes were randomized within blocks. After a retention interval of 1000 ms, a probe array of colored squares was shown and participants had to indicate whether all the colors of the probe array's targets were identical to the memory array or whether one of the squares had changed in color. I only considered subsegments of the time segment starting with the presentation of the memory array and ending 1000 ms after that for my analysis (see Section 5.1.2). The presentation of each block started with 12 practice trials. Then, each participant completed 360 trials of varying set size. Set sizes were randomized within blocks. After each block, participants got feedback about the accuracy of their responses. Given that Sander et al. (2012) assumed older adults and children would have additional difficulties with a cued hemifield presentation, they always presented the cue for 500 ms and showed it until the memory array was presented to minimize cue-related memory load. The cue direction was also blocked for 30 consecutive trials to prevent a task-switching situation that could differentially affect the age groups (e.g., Kray & Lindenberger, 2000).

5.2.2. Preprocessing

For preprocessing, the EEG was re-referenced to the two mastoid channels. Afterwards, an ICA was used to correct for remaining eye blink, noise, and muscle activity (Jung et al., 2000). Independent components representing artifactual sources were visually identified and removed from the data. Thereafter, trials with an incorrect response were removed. As the last preprocessing step PCA was used to project the EEG onto the principal components that explained 99% of the variance. This was done to restore the invertibility of the mean between-channel covariance matrix, which was violated by the removal of independent components in the previous preprocessing step. As explained above, the CSP method that I chose as part of the candidate models (see Section 5.1.2) requires an invertible mean between-channel covariance matrix. An almost identical but more elegant way to restore invertibility would have been to simply project the EEG onto the retained independent components. We (Karch et al., 2015) chose the ICA, PCA combination for practical reasons. As the exact location of individual frequency bands may change across the lifespan (Klimesch, 1999), the individual alpha peak frequency was estimated for each individual participant based on independently assessed resting state data. To determine the individual alpha peak frequency we (Karch et al., 2015) computed power spectra for eyes-closed resting state data and averaged them across all occipito-parietal electrodes. The individual alpha peak frequency was then defined as the maximum peak of the averaged power spectra between 7 and 13 Hz (see Sander et al., 2012, for more details). The cut-off frequencies for band-pass filtering into the alpha frequency ranges were determined in relation to individual alpha frequency based on suggestions by Doppelmayr, Klimesch, Pachinger, and Ripper (1998).

5.2.3. Data Analysis

Previous studies have examined load modulations of lateralized alpha power activity at 100 ms presentation times (Sauseng et al., 2009). Therefore, the analyses presented in this study are focused on this presentation time condition, which is the standard condition used in change detection paradigms (e.g., Luck & Vogel, 1997; Vogel & Machizawa, 2004). Analyses were conducted with custom-made MATLAB code based on the Fieldtrip software package (Oostenveld, Fries, Maris, & Schoffelen, 2010).

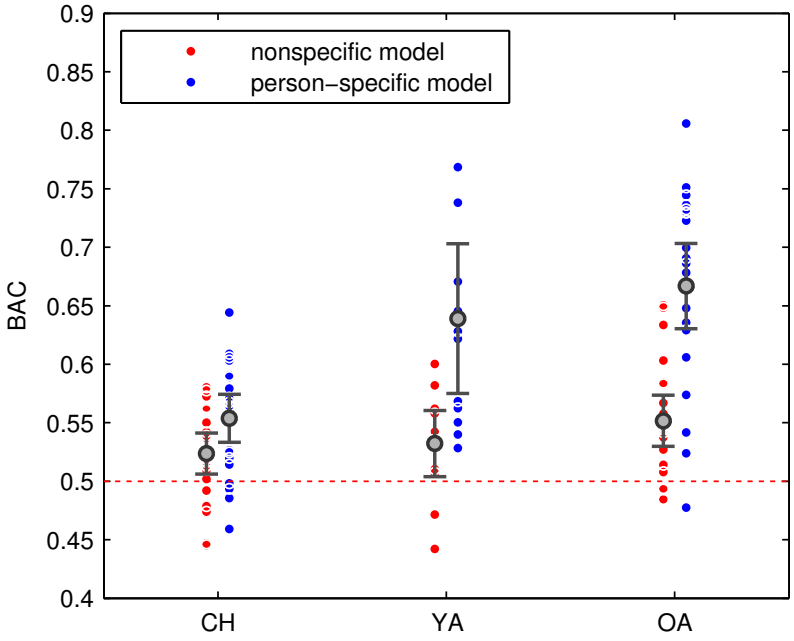
5.3. Results

First, I validate my framework and the set of candidate models by comparisons against chance and a theory-driven nonspecific model class. After that, I present detailed person-specific results for selected participants, followed by group results. The set of candidate models is comparatively flexible, making the results hard to interpret. Thus, I close this section with an evaluation of the impact of reducing the flexibility in order to arrive at potentially simpler models.

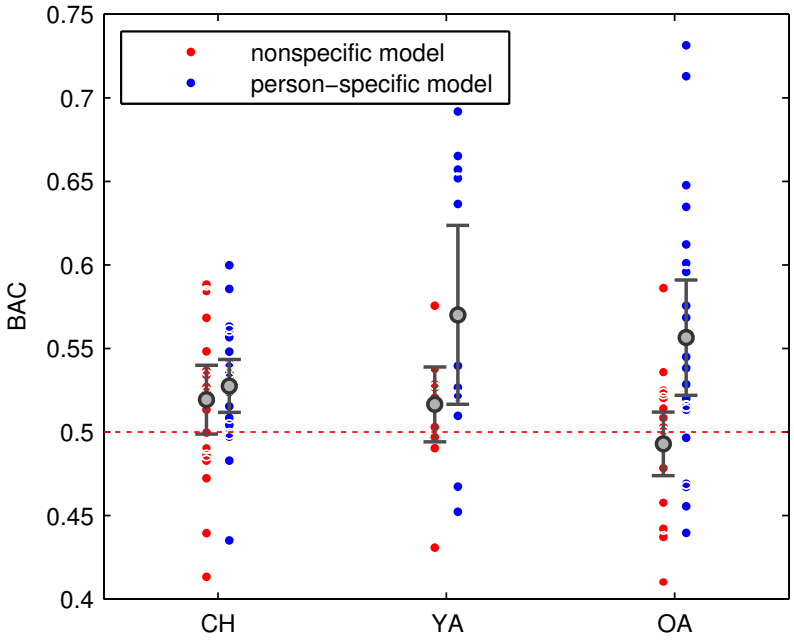
5.3.1. Performance Evaluation Against Chance and the Best Nonspecific Model

To validate my framework and the set of candidate models, I will demonstrate that my approach resulted in a classification performance significantly different from chance in each age group. Moreover, I will show that the accuracy is higher than for a conventional theory-driven model.

To test my approach against chance, I compared it against a model that guesses class membership on each trial. Such a decision function guessing for each trial will, in the limit, achieve a BAC of 0.5. Therefore, I concluded that the prediction accuracy of the person-specific models within an age group was reliably different from chance if the respective 95% CI of the mean BAC did not include 0.5. I calculated CIs based on the t-distribution, for each age group. To test for univariate normality of BACs within each group the Shapiro-Wilk test (Royston, 1995) and quantile-quantile plots were used. For all CIs the p-values for the Shapiro-Wilk test were larger than 0.05. Visual inspection of the quantile–quantile plots also suggested that the data were normally distributed. Furthermore, ceiling effects, typical for accuracy data, were not present. I therefore believe that the CIs based on the t-distribution were a reasonable choice for quantifying the reliability of my estimates. I conclude that the prediction accuracy of the person-specific models within an age group is reliably different from chance if the p-value of the one-sided t-test lies below 0.05. As Figure 5.4 shows for all age group–task combinations the CIs of the person-specific models did not include 0.5. Consequently, the p-value of the one-sided t-test was below 0.05 for all age groups. Hence, the person-specific models allowed for a reliable classification of attentional foci and WM load based on neural activity in the alpha frequency range. As effect size I report the difference between the mean BAC and 0.5. The effect sizes for children, younger adults, older adults



(a) Attentional Focus



(b) WM Load

Figure 5.4.: Mean balanced accuracy (BAC) and CI for the person-specific (blue dots) and the nonspecific model (red dots). Shown for (a) the attentional focus, (b) and working memory (WM) load classification for each of the three age groups separately. Each dot describes the BAC of one person. The box plots denote the respective mean values and 95% CIs . The red dotted horizontal line describes the BAC as expected by the null hypothesis. "CH" stands for children, "YA" for younger adults, and "OA" for older adults.

respectively were: attentional focus prediction: 0.0538, 0.1390, 0.1669; and WM load prediction: 0.0276, 0.0701, 0.0565.

To compare my person-specific approach against conventional theory-driven, nonspecific analysis schemes that assume the same model for each person, I needed an appropriate a-priori model. Previous studies have observed effects of shifted attention in early time windows (e.g., Freunberger et al., 2008), whereas effects of load were usually reported from the maintenance period of the change–detection task (Grimault et al., 2009; Vogel & Machizawa, 2004). Thus, I used 0–400 ms as the theory-driven time window for attentional focus prediction and 400–1000 ms for WM load prediction. In line with conventional analysis schemes (e.g., Pfurtscheller & Aranibar, 1977), the variances of the bandpass-filtered signals were used as features for the prediction instead of CSP. Conventional analysis techniques are univariate in nature. Therefore, the equivalent of a weight vector (or any other function that integrates evidence), as needed for linear prediction, is not available. Therefore, I used LLDA to learn the weight vector and classify the power features. I again used 10-fold cross-validation to evaluate its performance. To obtain the single best weight vector I applied cross-validation to the data of all participants simultaneously, but separately for both tasks. That is, I treated the data of all persons as a single person. As a result, I obtained an unbiased BAC estimate for the best estimated nonspecific model, explicitly implementing the expectation that the same neural mechanisms are present in all participants (Danziger, 1990; Molenaar & Campbell, 2009; Nesselroade, Gerstorf, Hardy, & Ram, 2007).

First, to examine whether there is an effect across all age groups of using the nonspecific or the person-specific model, I performed a repeated measures ANOVA with age group as the between factor and model type (person-specific or nonspecific) as the within factor. For the attentional focus prediction the main effect of model ($F(1, 53) = 61.44, p = 2 \times 10^{-10}$) as well as the interaction effect of model and age group were significant ($F(2, 53) = 7.62, p = 0.0012$). The interaction was ordinal, thus just revealing differences in the strength of the main effect between age groups. The main effect of model was positive. Hence, across all age groups, the person-specific model improved the BAC for the attentional focus prediction. For the WM load prediction the main effect of model ($F(1, 53) = 14.80, p = 0.0003$) was significant. The interaction effect of model and age missed the conventional significance level ($F(2, 53) = 3.01, p = 0.0577$). Hence, the positive main effect of model is interpretable. Thus, across all age groups, the person-specific model improved the BAC for the WM load prediction.

Figure 5.4 depicts the mean BAC for the nonspecific as well as the person-specific model. With regard to the classification of the attentional focus, for all three age groups the person-specific model was more accurate than the nonspecific model. The respective 95% CIs, effect sizes (mean difference), and p-values obtained by the paired one-sided t-test are shown in Table 5.1a. Similar results were obtained for the WM load prediction. Here, with exception of the children, the person-specific model was more accurate than the nonspecific model. The respective 95% CIs, effect sizes (mean difference), and p-values obtained by the paired one-sided t-test are shown in Table 5.1b.

Cross-validation results are not necessarily well approximated by a normal distribu-

	Nonspecific	Person-Specific	p-value	Mean Difference
Children	$[0.5061, 0.5412]$	$[0.5332, 0.5744]$	0.0258	0.0302
Younger adults	$[0.5040, 0.5605]$	$[0.5749, 0.7031]$	0.0018	0.1068
Older adults	$[0.5298, 0.5735]$	$[0.6305, 0.7033]$	9×10^{-8}	0.1152

(a) Attentional focus

	Nonspecific	Person-Specific	p-value	Mean Difference
Children	$[0.4988, 0.5400]$	$[0.5117, 0.5435]$	0.2520	0.0082
Younger adults	$[0.4942, 0.5390]$	$[0.5166, 0.6236]$	0.0313	0.0535
Older adults	$[0.4739, 0.5119]$	$[0.5220, 0.5910]$	0.0015	0.0636

(b) WM load

Table 5.1.: Mean balanced accuracy (BAC) and CI for the person-specific and the non-specific model for (a) attentional focus and (b) WMs load. Corresponding p-values from the paired one-sided t-test. These CIs are visualized in Figure 5.4

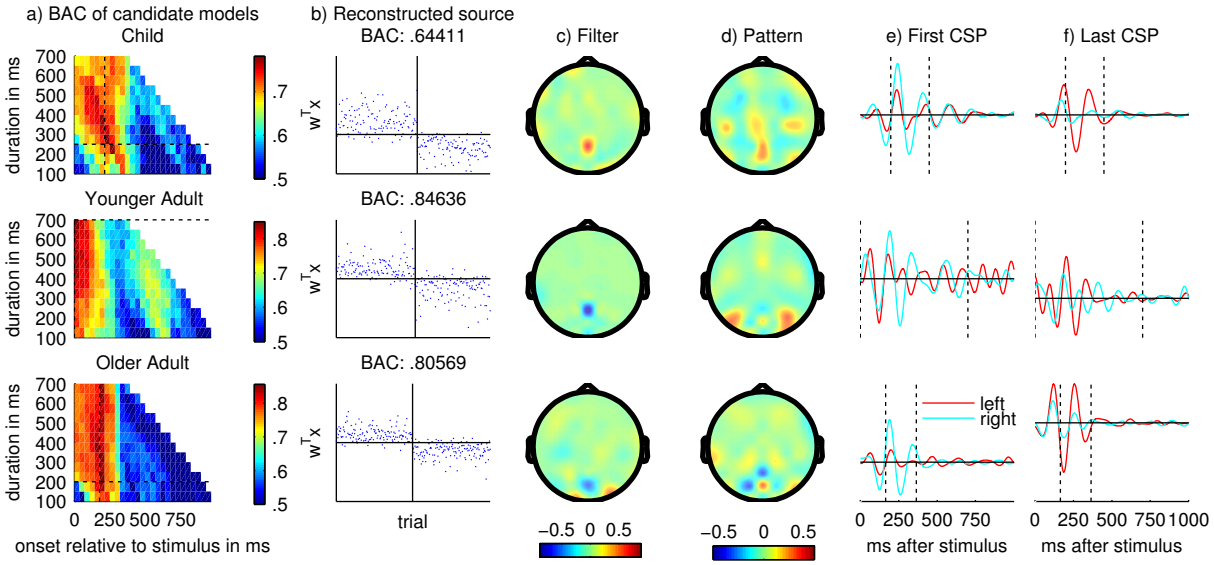
tion, and the variances for two classification methods are different depending on the means of the two distributions. Hence, a t-test is only an approximation of a valid test – in both cases, the test against chance and the comparison of the models against each other. However, in here the large effect sizes justify the t-test. To substantiate the test further, I repeated the test with a permutation test (Nettleton & Doerge, 2000) for the case with the smallest effect size (WM load prediction in children with person-specific model vs chance), confirming my results that even this effect size is significant ($p = 0.02$).

Overall the previous comparison clearly demonstrates that on average person-specific models were more accurate than the conventional approach, which assumes that the same neural mechanisms operate similarly in each person, a prediction captured in the nonspecific model. Moreover, for both tasks and all age groups, performance was better than random guesses for each trial.

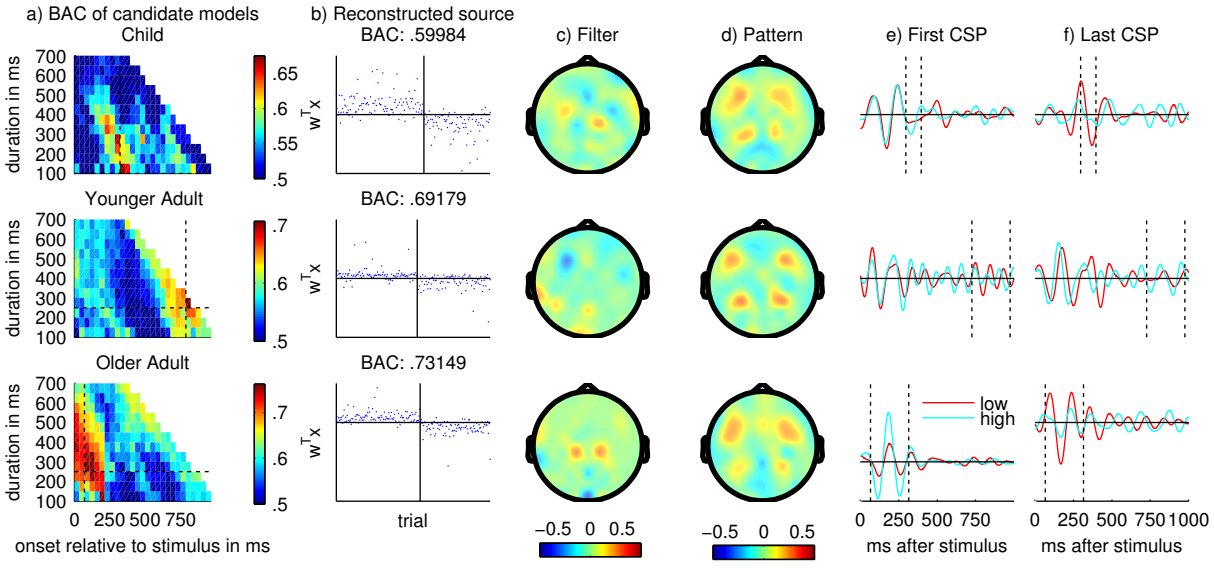
5.3.2. Person-Specific Results

Now that I have established that the person-specific models were more accurate than chance and the nonspecific model, the question is how to interpret the resulting estimated models both on an individual and on a group level. In the following I will describe how I interpreted the person-specific models and how I summarized the individual results on the group level.

In Appendix B, I show the person-specific results for all persons in both tasks. In Figure 5.5, I show the results for the person with the most accurate estimated model in each age group and task. The properties of the selected person-specific models quantify different aspects of the observed data. Each candidate model was derived from the



(a) Attentional focus



(b) WM load

Figure 5.5.

Figure 5.5.: Person-specific results. Examples of results for a) attentional focus and b) working memory (WM) load prediction for single individuals, one from each age group. Column (a) shows the estimated balanced accuracy (BAC) for the different candidate models. The x-axis describes the onset and the y-axis the duration of the corresponding time window. Colors refer to the estimated BAC of the respective candidate model, with hot colors indicating higher BAC and cold colors lower BAC. The crosshair depicts the location of the selected model. Column (b) shows the estimated source component for each trial as reconstructed by the best estimated model, including the BAC of the best model. The trials are sorted by their true class. The vertical line separates the classes. The horizontal line marks 0. For a perfect classifier the estimated source must be positive for all trials to the left and negative for all trials to the right of the vertical line. Column (c) shows the entries of the normalized filters. Column (d) shows the normalized pattern. Column (e) shows the mean time series for the first common spatial pattern (CSP) filter for both classes. Column (f) shows the mean time series for the last CSP filter for both classes. The x-axis describes the time elapsed since the onset of the memory array. The vertical dotted lines indicate the selected time window. The horizontal line in columns e and f) marks 0.

same basic class, but they vary with regard to exposure to certain spatio-temporal features of the observed brain responses. The onset and duration of time windows each model is exposed to precisely describe the temporal information used by a given model. Within the spatial domain, the resulting spatial filter and pattern coefficients reflect the topographical information exploited by the chosen model.

The BACs estimated for the different candidate models (column a in Figure 5.5) map out a space of information content across different choices of time windows. They also quantify the uncertainty of the model selection procedure. If there is only one model with a high BAC, one can be relatively certain that the truly best model was selected. If there is almost no difference between the model with the highest BAC and several other models, however, model selection is mostly determined by random variation. I only report the BACs of the candidate models that have the same number of CSP filter pairs as the most accurate candidate model. This is motivated by the fact that we (Karch et al., 2015) were mostly interested in the location of the time window. In addition, the number of CSP filter pairs was the least critical setting by which the candidate models differed (see Section 5.3.4). Furthermore, I report the following properties of the person-specific models: the BAC estimate as obtained by nested cross-validation, the classifier output (= reconstructed source) $w^\top x_t + c$ for each trial t sorted by true class membership as a visualization of the predictive behavior (both column [b]), the filter (column [c]) and the pattern (column [d]) to describe the spatial information that was employed by the model, the selected time window to quantify which time segment was employed for prediction, and the mean (over trials) time series of the first (column [e])

and last (column [f]) CSP component for each class as a visualization of the time course of the most discriminative projected components.

I will exemplarily describe the results for the younger adult in the attention task (second row in Figure 5.5). The grid of candidate model BACs shows that candidate models employing time windows starting early have a high BAC (column [a]), while none starting later than 200 ms achieves a comparable BAC. The reconstructed source shows the nice separability of the two conditions for this person, which is also reflected in a BAC of 0.8464 (column [b]). The filter has a high negative value at Pz, indicating that higher power at this electrode is evidence for the "Attention right" condition (column [c]). The pattern has high positive values in parieto-occipital areas, indicating that if the decision function predicts "Attention Left", these areas show higher power (column [d]). The longest possible time window was selected (700 ms) and starts directly after stimulus onset. The CSP components both show the desired effect within the chosen time window: increased variance for one class and decreased variance for the other class. This effect is not present outside the selected time window (columns [e] and [f]).

5.3.3. Group Results

Figures 5.6 and 5.7 show the group results. The scatter plot of onsets and durations of the selected time windows (Figures 5.6a and 5.7a) illustrate their between-persons variability. The mean (across persons) BAC for the different candidate models is shown in Figures. 5.6b and 5.7b. As for the person-level results, we (Karch et al., 2015) were most interested in the location of the time window, so I only report the mean by onset and duration of the employed time window. To achieve this I averaged across the different numbers of CSP filter pairs within a person prior to averaging across all persons. The mean BAC for the different candidate models map out a space of mean information content across different choices of time windows.

Figures 5.6c–e and Figures 5.7c–e show three time series. The first expresses how often each time point was part of the selected model (Figures 5.6c and 5.7c respectively). For each time point, I counted the number of persons for whom this time point was included in the person-specific model and then divided this by the total number of persons in this group. This is a way of visualizing the information per time point. However, as it only relies on the best estimated model it ignores the useful information of the candidate models that were not selected. Therefore, for a second time series I compressed the mean BAC of the candidate models in a similar fashion by calculating the mean BAC for every time point (Figures 5.6d and 5.7d). I did this by averaging the mean BACs of all estimated models that contain the respective time point. For models that employ a very long-lasting time window, however, this could be misleading as the high performance may be driven by a localized part. To remedy this, I also report the mean BAC for every time point but limited to the candidate models based on the shortest time window (100 ms) (Figures 5.6e and 5.7e). I will refer to these results as coarse (Figures 5.6d and 5.7d) and fine mean (Figures 5.6e and 5.7e) BAC by time.

I do not report any averages of the person-specific filters and pattern, as they originate from different time windows and thus, their interpretation is difficult.

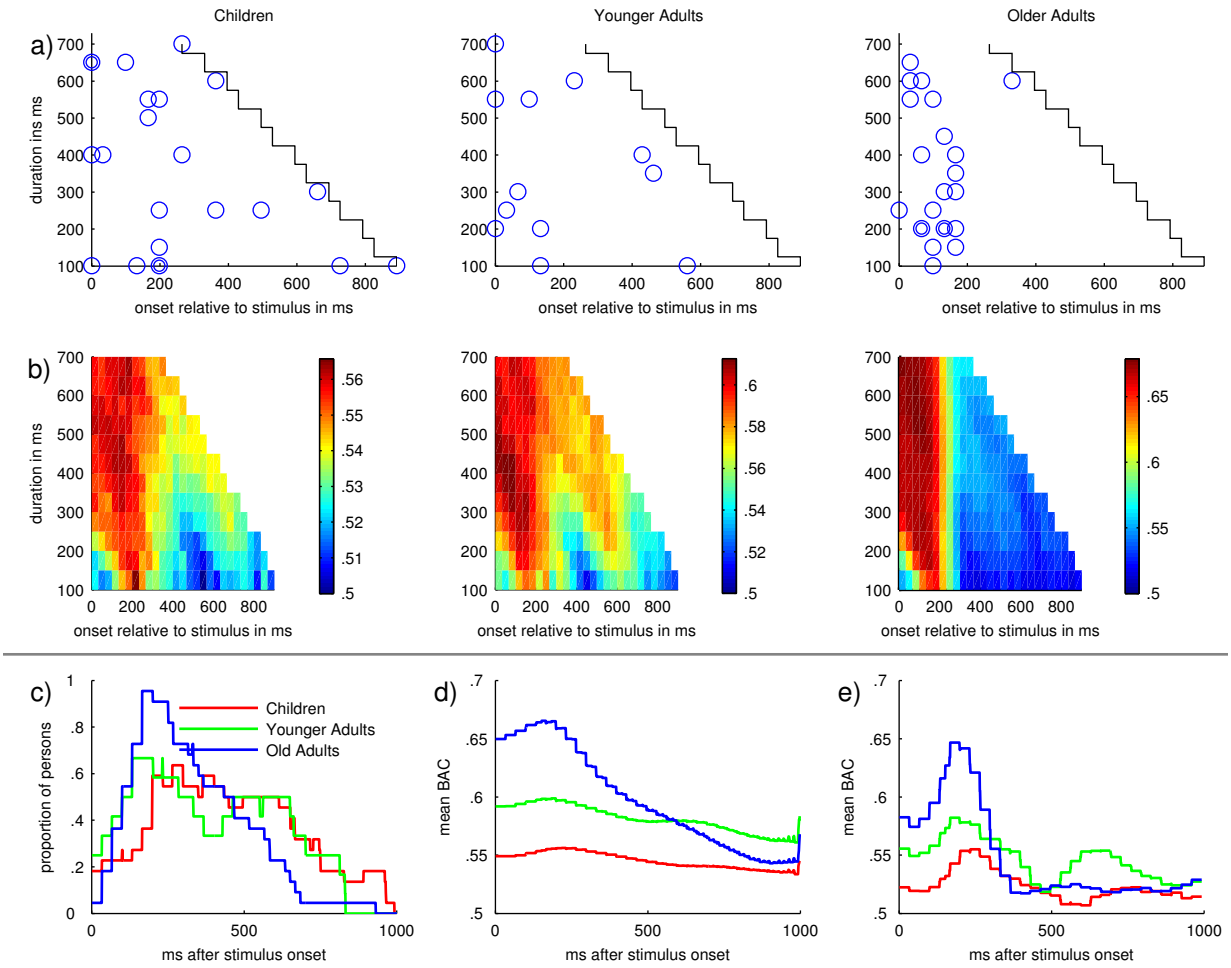


Figure 5.6.: Group results for the attentional foci prediction. In row (a) and (b), each column shows the results for one age group. In row (a) each point represents the corresponding time window of one person-specific model. Double rings indicate that the corresponding time window was selected for two persons. The x-axis describes the onset and the y-axis the duration of the selected model. The step function encloses the region of possible time windows. In row (b) the mean BAC for the candidate models by onset (x-axis) and duration (y-axis) are shown. Colors refer to the estimated BAC of one candidate model, with hot colors referring to higher BAC and cold colors to lower BAC. Panel (c) shows for each time point the proportion of persons whose model used the respective time point, for each age group. Panel (d) shows the mean BAC by time using all candidate models (coarse mean BAC by time). Panel (e) shows the mean BAC by time using only the candidate models with a time window lasting 100 ms (fine mean BAC by time).

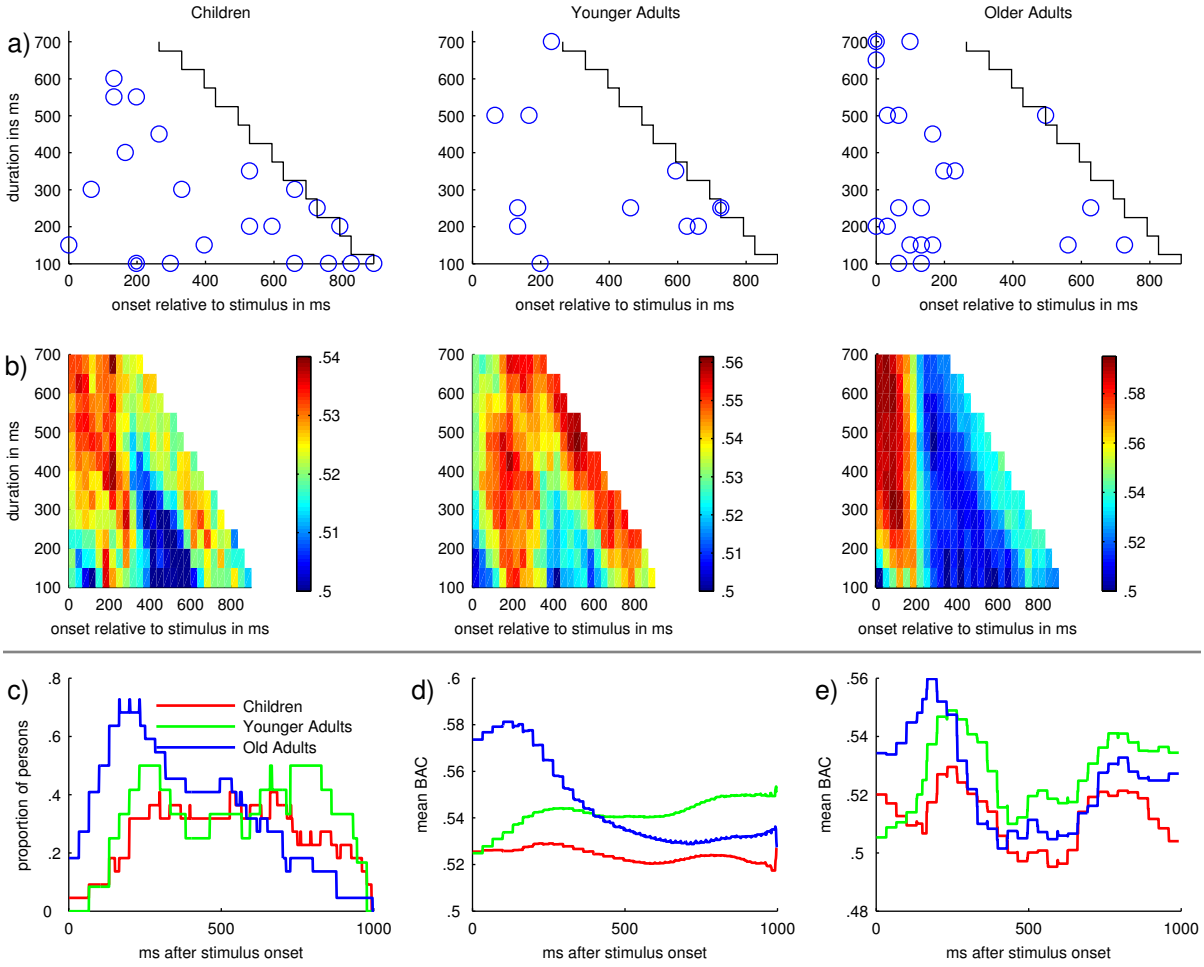


Figure 5.7.: Group results for the WM load prediction. For a description see Figure 5.6.

Attentional Focus

Figure 5.4a showed that mean BAC was highest for the older adults (0.6669), followed by the younger adults (0.6390), and lowest for the children (0.5538) for attentional focus. Figure 5.6 shows the group results.

For the children, the employed time window of the best estimated model varied across the whole range of durations and onsets (Figure 5.6a). This finding may indicate larger variability in the group of children compared to the other age groups. However, given that the mean BAC was only slightly higher than chance level, this must be interpreted with caution.

For the majority of younger adults, models that start earlier than 250 ms after memory array onset were selected (Figure 5.6a). For the remaining three younger adults, models starting later than 400 ms were picked. No model beginning later than 600 ms was selected. However, some of the early models contain data for almost the complete trial because of their duration (see Appendix B). The fine mean BAC by time plot (Figure 5.6e) shows two distinctive peaks around 200 ms and around 700 ms for the younger adults. This is a consequence of the fact that in the group of younger adults, I observed two clusters: For one cluster, an early time window provided good discriminability, and for the other, a later time window discriminated well. This is reflected by the fact that the BAC for the candidate models peaked around the best model and was relatively low for the remaining candidate models for every younger adult (see Appendix B).

The distribution of time windows in the older adults is strongly shifted to the stimulus onset. For only one older adult a model starting later than 200 ms post-stimulus was selected, with a relatively low BAC of 0.5417 (Figure 5.6a). The best estimated model contains data after 700 ms post-stimulus for only one person, while almost all person-specific models contain data around 200 ms post-stimulus. The coarse (Figure 5.6d) and fine (Figure 5.6e) mean BAC by time plots clearly show an early peak (at roughly 200 ms) followed by a sudden decline. This indicates that for almost all older adults, discriminability was best if early time windows were employed. It is noteworthy that the accuracy of these models was superior to those of the younger adults (see Figure 5.4a).

Working Memory Load

The mean BAC for WM load was highest for the younger adults (0.5701), followed by the older adults (0.5565), and lowest for the children (0.5276) (see Figure 5.4b). Figure 5.7 shows the group results for the WM load prediction.

For the children, the employed time window of the best estimated model varied across the whole range of duration and onset (Figure 5.7a). Again, this finding may indicate larger variability in the group of children compared to the other age groups. However, given that the mean BAC was only slightly higher than chance level, this finding must be treated with caution. For many of the younger adults and the older adults (6 of 12 younger adults and 18 of 22 older adults), models that started earlier than 250 ms after memory array onset were chosen (Figure 5.7a). After a gap, there is another group of people with models that start after 450 ms (6 younger adults and 4 older adults).

Note that the group of persons for whom early models were picked is much larger in the older than in the younger adults. Some of the early models contain data for almost the complete trial because of their duration.

Accordingly, the mean BAC for the candidate models shows two peaks for the younger and the older adults (Figure 5.7b). One represents early models and the other late models. For the younger adults, both peaks are equally strong. For them, the coarse BAC by time plot (Figure 5.7d) is therefore almost constant. Consequently, their fine BAC by time plot (Figure 5.7e) shows an early and a late peak.

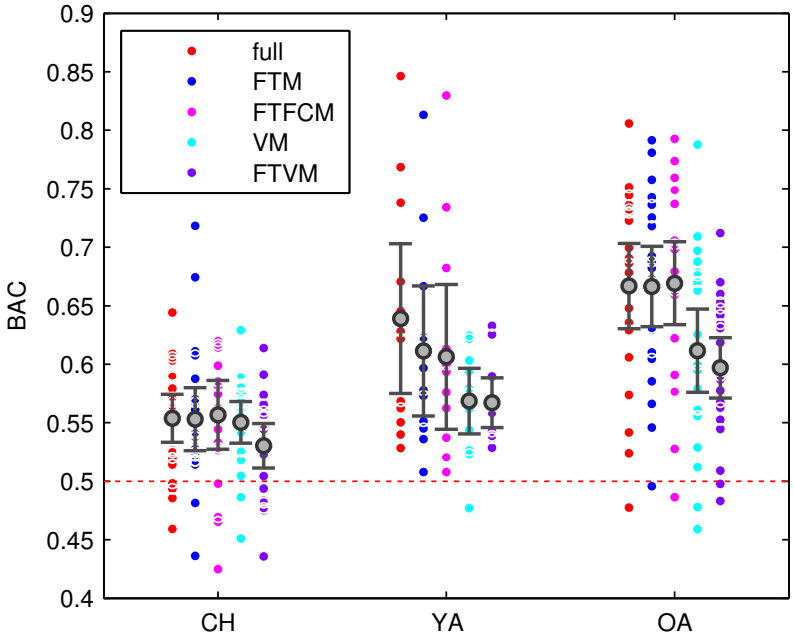
Despite the fact that there is a peak for late models, the coarse BAC by time plot (Figure 5.7d) for the older adults is qualitatively the same as for the attentional focus prediction: An early peak around 200 ms and then a sudden drop. The fine mean BAC by time reveals both peaks.

For both age groups, the reasons for the late and the early peak are less clear than for the attentional focus prediction. There are persons for whom only early models were accurate, persons for whom only late models were accurate, and persons for whom early and late models were accurate (see Appendix B).

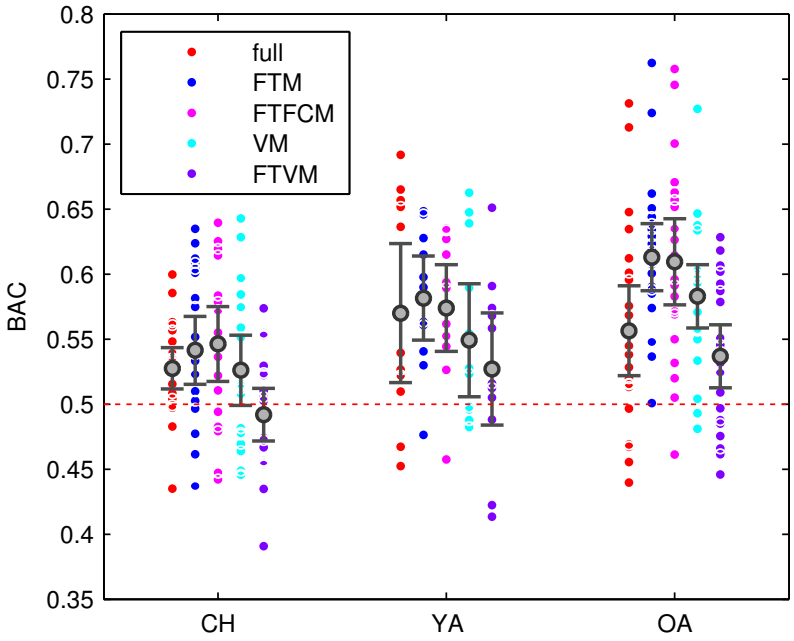
5.3.4. Performance Comparison Against Simpler Person-Specific Models

The set of candidate models that I proposed in order to obtain person-specific models is comparatively flexible (i.e., it adapts many parameters to the data) and uses sophisticated algorithms: model estimation entails parameter selection over time windows, a spatial projection, and a subsequent estimation of regularized regression weights. As a consequence results are hard to interpret. In the following, I report the results of an evaluation of the impact of reducing parameters to arrive at potentially simpler models. To this end, I gradually reduced the complexity of the candidate models. All the p-values that I report in this section were calculated using a two-sided paired t-test, as I hypothesized that the simpler model classes would be either better or worse then my original model class. At first sight it might seem unreasonable to hypothesize that a simpler model class leads to better predictive performance, as more complex model classes always lead to a better model fit. However, a better model fit does not necessarily lead to better predictive performance. This is due to the fact that a more flexible model class is more vulnerable to noise in the data. This is called overfitting in the literature (Bishop, 2006; Duda et al., 2001; Hastie et al., 2001).

In a first step, I abandoned the optimization across the time window by fixing it to a theory-driven estimate for each task (0–400 ms for attentional focus, 400–1000 ms for WM load). I call this model class the *fixed time model* in the following. By fixing the time windows I reduced the number of candidate models from 1220 (244 time windows by up to 5 components) to 5. For the attentional focus prediction, the mean BAC of the fixed time model was not significantly different from the mean BAC of the full model class for both the children ($p = 0.9605$) and the older adults ($p = 0.9265$). The mean BAC of the full model was slightly higher for the younger adults ($p = 0.1455$). The results for the older and younger adults are not surprising considering my previous analysis: For almost every older adult, a model with an early component was accurate.



(a) Attentional focus



(b) WM load

Figure 5.8.: Mean BAC and CI for the five different model classes for each of the three age groups within the attentional focus (a) and WM load prediction tasks (b). Each dot describes the BAC of one person. The box plots denote the respective mean values and 95% CI intervals. The red horizontal line describes the BAC as expected by the null hypothesis. "CH" stands for children, "YA" for younger adults, and "OA" for older adults. "FTM" stands for "fixed time model," "FTFCM" for "fixed time fixed CSP model,", "VM" for "variance model," and "FTVM" for "fixed time variance model" (see text for details).

For the younger adults, time window optimization improved the results as there were two clusters: one for which an early, and one for which a late component is most predictive of the behavioral difference. For the children this again suggests that the observation of large variability might in fact be due to noise.

Regarding the WM load prediction, the mean BAC of the fixed time model was not significantly different from the mean BAC of the full model class for both the children ($p = 0.3465$) and the younger adults ($p = 0.5731$). For the older adults, the mean BAC of the fixed time model was higher than the mean BAC of the full model ($p = 0.0013$).

As a further step of model simplification, I abandoned the optimization of the number of CSP filter pairs. Instead, I picked the first three filter pairs. This was previously found to yield good predictive performance (Blankertz et al., 2008). I refer to this model as the *fixed time fixed CSP model*. Across all age groups and tasks, the additional restriction of the model space did not significantly influence the accuracy in comparison to the fixed time model (attentional focus: children: $p = 0.6075$, younger adults: $p = 0.4311$, and older adults: $p = 0.5450$; WM load: children: $p = 0.5600$, younger adults: $p = 0.4184$, and older adults: $p = 0.6980$). This is further evidence that three filters represent a reasonable choice.

To judge if CSP is a reasonable feature extraction method for the present data, I introduced two more model classes that do not rely on CSP features but directly operate on the variance of each channel. The two classes differed from each other in the following: In one class, I optimized the time window as before. I call this model class *variance model* in the following. I compared this class against the full model. For the other class, I fixed the time window to my theory-driven estimates (0–400 ms for attention, 400–1000 ms for load). I call this model class *fixed time variance model*. I compared this class against the fixed time model. For the attentional focus prediction the variance model was less accurate than the full model for both the younger and the older adults ($p = 0.0256$ and $p = 0.0004$). For the children there was no significant difference ($p = 0.7986$). For the WM load prediction there was no significant difference between the variance model and the full model for any age group (children: $p = 0.9193$, younger adults: $p = 0.2728$, and older adults: $p = 0.1355$). The fixed time model was more accurate than the fixed time variance model for both tasks and all age groups (attentional focus: children: $p = 0.0902$, younger adults: $p = 0.0533$, and older adults: $p < 0.0001$; WM load: children: $p = 0.0019$, younger adults: $p = 0.0382$, and older adults: $p = 0.0007$). Thus, for most comparisons, CSP features led to more accurate models. However, especially for WM load prediction for the older adults the combination of CSP and time selection seems to lead to overfitting.

The discussion of the results will be solely based on the full model. However, from a classification perspective, the full model was only as accurate as a simpler model class using fixed time windows (fixed time model) in conjunction with CSP and LDA. These time windows were derived from the literature and were constrained to an early time window for the classification of the attentional focus and to a later time window for the classification of the WM load. Using these fixed windows, with exception of the WM load prediction in the older adults, I almost achieved the same BACs as with time

window optimization. Hence, two models, the full model and the fixed time model, do not differ significantly in predictive accuracy. In other words, the hypothesis that their performance on a new observation is equal can not be rejected. It is generally taken for granted that more parsimonious explanations for a set of observations are preferable under comparable goodness-of-fit (Akaike, 1998). In my case the more parsimonious explanation corresponds to the fixed time model, as it is a special case of the full model. But, since I relied on cross-validation to estimate the predictive accuracy, I implicitly accounted for the unequal complexity between the models in terms of numbers of free parameters. Hence, unless model fit was evaluated directly for new observations, choosing the simpler model may be a good heuristic – or it may not. Insofar, I allow myself to draw conclusions from the two hypotheses in head-to-head competition. In other words, the full model provides evidence for interindividual variability in brain responses, which would have remained covered by the fixed time model analysis alone. Hence, the models based on the full model, although not superior with regard to prediction accuracy, provide additional insights about reliable idiosyncrasies in brain–behavior mappings useful for further theory-building. Interestingly, for older adults the BAC achieved by my time-optimizing model was even worse than that of the fixed time model. One possible explanation may be that for the WM load prediction, the discriminating signal is too weak so that my complex model class, containing many candidate models, resulted in overfitting of the data. That is, the model selection process is disturbed by the noise in the data such that a theory-driven model selection leads to more accurate predictions.

6. Summary and Discussion

Repeated measures obtained from multiple individuals are of crucial importance for developmental research. To analyze such data, a model for inter-individual as well as intra-individual variation is required. In this thesis, I developed two new modeling approaches for repeated measures obtained from multiple individuals: (1) GPPM, a new panel modeling method based on the flexible function fitting approach GPR, and (2) a method to obtain and summarize person-specific models for EEG data based on machine learning techniques.

6.1. Gaussian Process Panel Modeling

The motivation for developing GPPM was the observation that panel data are typically modeled with longitudinal SEM or a special case thereof (like HLM). While using SEM is a viable option for modeling panel data, it also possesses multiple weaknesses. First and foremost, longitudinal SEMs are inherently discrete-time models and, thus, model intra-individual variation inappropriately, since the observed intra-individual variation is typically the result of a continuous-time process. Also, due to its restriction to linear equations, SEM can only express a relatively limited number of models of intra-individual variation.

To develop GPPM, I extended the Bayesian continuous-time time series modeling method Gaussian process time series modeling (GPTSM) to be applicable to multiple independent time series, i.e., a panel data set (see Chapter 3). I showed that the resulting method can incorporate a large set of models for the intra-individual and inter-individual variation. Besides introducing the Bayesian inference procedure used for GPTSM for GPPM, I also developed frequentist inference procedures for GPPM because this connects to conventional data-analysis procedures and merely modifies the range of models available for applied psychologists without requiring a change of inference approach. For model selection, that is, selecting between a set of different GPPMs, I proposed using cross-validation, as popularized in machine learning, as there is evidence that it is in general favorable over other approaches (Kearns et al., 1997). I also showed how person-specific predictions can be obtained using GPPM.

To further illustrate the strengths and weaknesses of GPPM, I compared it to existing panel methods, particularly longitudinal SEM and multiple-subject SSM (see Section 4.1). I showed that both longitudinal SEM and multiple-subject SSM can be regarded as special cases of GPPM. At the same time, GPPM is able to express more models than the existing panel methods.

The main technical difference between longitudinal SEM and GPPM, besides the latter

being more general, lies in the way how the model-implied moments (the mean and the covariance matrix) are specified. In longitudinal SEM, the model-implied moments are implicitly described via linear structural equations, whereas they are explicitly described via mean and covariance functions in GPPM. This rather technical difference leads to multiple practical differences. First and foremost, in contrast to longitudinal SEM, GPPM is well suited for continuous-time modeling. In line with previous work (Oud & Singer, 2008; Voelkle et al., 2012), I argue that adapting continuous-time modeling for the analysis of panel data is among the most important methodological shifts that need to happen within developmental psychology. Using discrete-time modeling results in introduction of unnecessary assumptions, which are never fulfilled, that therefore lead to invalid inferences.

Another important advantage of GPPM over longitudinal SEM is that person-specific predictions can be obtained readily. This makes it possible to estimate person-specific trajectories that also take into account the data of all other persons. Person-specific predictions can, for example, be used as screening devices for interventions by identifying those persons who are at risk of developing into a unsatisfactory direction. I exemplarily demonstrated the versatility of GPPM as compared to longitudinal SEM by showing that it allows easy extension of a continuous-time LGCM by a continuous-time AR(1) error structure. Thus, GPPM addresses the two weakness of longitudinal SEM that I identified earlier. In contrast to SEM, GPPM is well suited for continuous-time modeling and it is able to express a richer family of models for intra-individual variation, as it is not restricted to models for intra-individual variation that can be expressed using linear equations.

Besides being more general, GPPM also has other advantages over multiple-subject SSM. First and foremost, the description of the model-implied moments is again distinct. Whereas multiple-subject SSM implicitly represents the model-implied moments of a GP using SDEs, GPPM delineates them directly. Thus, GPPM can be seen as a static (Hertzog & Nesselroade, 2003; Voelkle, in press) continuous-time modeling method, whereas multiple-subject SSM can be regarded as dynamic (Hertzog & Nesselroade, 2003; Voelkle, in press). Depending on the form of a researchers' theory, translating it into either a dynamic or a static model might be easier. In this way, GPPM and multiple-subject SSM allow complementary perspectives on the modeling of change. However, GPPM does not only offer an alternative perspective. GPPM also extends the range of models that multiple-subject SSM offers.

As an example for a GPPM that cannot be expressed using multiple-subject SSM or SEM, I presented the exponential squared model, which implements the "nature does not jump" assumption put forward by thinkers such as Darwin (1859) and Leibniz (1704/1886) (see Section 4.2.1). I showed that the exponential squared model is similar to the continuous-time AR(1) model, but in contrast to the AR(1) model it implies smooth trajectories. I presented the finding that the exponential squared model is selected over the AR(1) model using conventional model selection methods on a authoritarianism data set (Heitmeyer, 2004), which has previously been analyzed using the AR(1) model (Voelkle et al., 2012). Thus, I could show that the stability of authoritarianism is

better represented by the exponential squared model than by the common AR(1) model. This provides first empirical evidence that the exponential squared model is well worth considering for psychological research.

As an example for the versatility of GPPM, I showed how the error process in a LGCM can be easily replaced in its GPPM representation (see Section 4.2.2). Particularly, I presented the way that continuous-time AR(1)-correlated errors can be implemented. In parallel to the example introducing the exponential squared model, I demonstrated that the LGCM with AR(1)-correlated errors is selected over the regular LGCM with uncorrelated errors on the positive affect data from the COGITO study (Schmiedek et al., 2010). This demonstrates the flexibility of GPPM.

Thus, I was able to show that GPPM addresses multiple shortcomings and provides a different perspective as compared to the existing methods multiple-subject SEM and SSM. Accordingly, it represents a viable addition to the modeling toolbox. GPPM adds a new perspective on modeling with a broader class of models, and thereby should contribute to a broader application of continuous-time models in psychological data analysis. To the best of my knowledge, the proposed GPPM approach constitutes the first proposal for using GPs for the analysis of panel data within psychology. This thesis also contains the first unified perspective on SEM, multiple-subject SSM, and GPPM. Previous work compared the two methods that GPPM is based on and thus closely related to, namely GPR and GPTSM, with SSM (Hartikainen & Särkkä, 2010; Särkkä & Hartikainen, 2012), but there no work comparing SEM and GPR.

One weakness of GPPM in comparison to both SEM and multiple-subject SSM is that the concept of latent variables is not explicit. Extending GPPM in this vein should be straightforward and remains to be done.

A further weakness is that the specification of a GPPM for a multivariate time series is still cumbersome (see Section 3.4.2). To this end, I envision a combination of SEM and GPPM: GPPM is used to describe a model of change for potentially multiple latent variables and SEM is used to describe the measurement model, that is, how the latent variables are mapped onto the actual observations. In this vein, the strength of both methods could be leveraged. On the time dimension the model can be described using the powerful concept of a set of GPs used by GPPM. On the measurement model dimension, the restrictions of SEM are not as harmful, since the number of measurements per time point are finite. At the same time, the theory of constructing appropriate measurement models is well developed within the SEM community. Note that the resulting model is technically still a GPPM.

Since GPPM contains most conventional panel methods as special cases, it might also be useful for software development. Instead of developing and maintaining a separate software for each modeling approach (e.g., SSM, HLM, SEM), only the GPPM software would need to be maintained. For the specific approaches, only respective input and output interfaces would be required. The input interface would allow model specification in the customary language of each approach, and the output interface would likewise represent the results in the approach-specific fashion. Besides saving time, using the same core for all modeling approaches would allow more people to work on this core

and at the same time a larger user base would profit from improvements to it, in a clear betterment of the current situation, where independent software for each approach is developed.

The utility of GPPM as basis for a universal software package would be greatly enhanced if it were able to model non-Gaussian data, for example binary or count data. Indeed, an extension of GPPM to model non-Gaussian data is straightforward. In GPR, as for generalized linear models, so-called link functions (Rasmussen, 2006, Chapter 3 and Chapter 9.3) are used to accommodate non-Gaussian data. Using these slightly complicates parameter estimation. However, the appropriate algorithms have been developed for GPR and are included in the GPML toolbox (Rasmussen & Nickisch, 2015). Translating the relevant work to GPPM should be simple.

Another interesting direction for future work would be to use the established connections of GPR to multiple other machine learning methods, in order to further understanding of the connection between what seem to be very different methods at first sight. GPR is closely related to Bayesian regression (Rasmussen, 2006, Chapter 2). There is also work connecting GPR to many other methods from supervised machine learning (Rasmussen, 2006, Chapter 6) such as support vector machines. In Section 4.1.1, I presented the connection between GPPM, which is closely related to GPR, and SEM and SSM. Thus, GPPM provides a good starting point for understanding the differences and commonalities of many methods such as SEM, as used in psychology, and support vector machines, as used in machine learning.

Since GPPM is more general than SEM, I also investigated the question whether transforming SEMs into their equivalent GPPMs speeds up the time required to obtain parameter estimates (see Section 4.3). Speeding up parameter estimation for longitudinal SEM is of crucial importance, especially for data sets with many time points, which are increasingly collected in diary and experience sampling studies (Bolger & Laurenceau, 2013). My results suggest that GPPM software is indeed faster than SEM software. However, the obtained speedups were not as substantial as hoped for. For the AR(1) model, the speedup was large enough to recommend using GPPM instead of SEM. Yet, the AR(1) model can be fitted much faster by the Kalman Filter, using its multiple-subject SSM representation. For the LGCM the speed enhancement was not great enough to recommend using GPPM instead of SEM. However, translating a SEM to a GPPM has the potential to reduce the time needed for parameter estimation much more dramatically. Within the GPR community, there are already many different approximation algorithms designed to speedup parameter estimation at the cost of a slightly reduced accuracy (e.g., Csató, 2002; Freytag et al., 2012; Park & Choi, 2010; Särkkä & Hartikainen, 2012). Using these algorithms for the estimation of the parameters of a GPPM might dramatically reduce the time taken for parameter estimation. To what degree the reduced accuracy is tolerable in practice is also an issue for further research. Possibly, the most promising algorithm is the one proposed by Särkkä and Hartikainen (2012). They have proposed converting a GPR model into the equivalent SSM. This is not possible for every GPR model, but each one can be approximated. The parameter estimates of a SSM can be obtained linearly in time rather than to cubically

(x^3) for SEM and GPPM. Adapting the approximation algorithms developed for GPR for GPPM may thus have the potential to fit GPPMs and consequently longitudinal SEMs linearly in time. This would constitute a tremendous speedup compared to the current cubical running time.

6.2. Person-Specific EEG Modeling

The motivation for developing the approach to derive person-specific models for EEG data was based on the fact that increased behavioral and neuronal inter-individual variation in both children and older adults is observed in lifespan samples (Lindenberger, Burzynska, & Nagel, 2013; Nagel et al., 2009; Werkle-Bergner et al., 2012). At the same time conventional EEG analysis approaches ignore within-group inter-individual variation with the aforementioned consequence that the group model might not be representative for any person within the group.

To derive person-specific EEG–behavior mappings, or in other words, models, I developed an approach relying on machine learning methods (see Section 5.1). The general idea was to select the best model from a set of candidate models and at the same time obtain an unbiased estimate of the accuracy of the selected candidate model. The algorithm was tailored to problems in which all inferences are to be performed on typical EEG data sets, that is, the number of examples (trials) per person is small. Therefore, I proposed using a framework involving a nested cross-validation loop to allow for unbiased model selection and performance estimation.

To validate the proposed method I re-analyzed EEG data from a study that targeted brain oscillatory mechanisms for WM selection and maintenance in a lifespan sample including children, younger, and older adults (Sander et al., 2012). My aim was to find person-specific models that describe the mapping from the modulations of rhythmic neural alpha band activity to (a) the focus of spatial attentional and (b) the number of items in WM load.

To derive the person-specific models on the WM data set, I employed the combination of CSP and LDA as core techniques from which to derive a set of candidate models. This setup is popular for building BCIs based on rhythmic neural activity (e.g., Blankertz et al., 2008; Blankertz et al., 2011; Ramoser et al., 2000). The candidate models differed in terms of the spatio-temporal information they were exposed to. In terms of the temporal dimension, each candidate model was exposed to a particular time-window within each trial. In terms of the spatial dimension, an optimal spatial weighting was obtained by the combination of CSP and LDA.

My work is distinct from typical BCI research in that I did not aim to construct a real-time classification system but rather used the same methods to obtain person-specific models. Thus, in contrast to classical BCI research, the interpretation of the obtained mappings as person-specific models was important. I showed how the spatial information contained in the estimated models from the proposed model class can be meaningfully interpreted as patterns and filters, a usual way to summarize multivariate models for EEG data. Interpretation on the temporal dimension is straightforward, since every

model is exposed to a particular time window.

The results demonstrate the potential of the proposed approach to derive person-specific models (Nesselroade et al., 2007) in age-comparative EEG studies (see Section 5.3). WM load as well as the focus of spatial attention could be discriminated reliably in all three age groups based on rhythmic neural activity in the alpha frequency range. In all three age groups, the BAC of person-specific models provided a significantly above-chance classification for the prediction of both attentional focus and WM load. Also, across age groups and for both prediction targets, the person-specific models were more accurate than a theory-driven nonspecific model disregarding inter-individual variation, with the exception of the children's model for the prediction of WM load. Hence, the present framework demonstrates the feasibility of deriving person-specific models based on EEG data.

Usually, the spatio-temporal differences in the neural responses between persons are assumed to reflect measurement error (e.g., Luck, 2005). However, my approach shows that individually-tailored models do not just fit noise, as exemplified by above-chance BAC derived from nested cross-validation. Rather, in terms of prediction accuracy, they outperform conventional nonspecific models that assume no within-group inter-individual variation (e.g., Danziger, 1990). This finding underscores the idiosyncrasy of neural responses underlying similar overt behaviors, and calls for further studies to investigate how this is related to inter-individual variation both on the neural level (e.g., in terms of differences in brain structure; cf. Breakspear, Jirsa, & Deco, 2010; Deco, Senden, & Jirsa, 2012; Valdes-Hernandez et al., 2010) and on the behavioral level (e.g., in terms of differences in strategy use; cf. Corbin & Marquer, 2009).

One of the major challenges when dealing with person-specific models is how to extract and aggregate the person-specific information across individuals in order to allow group comparisons. Here, I suggest summarizing person-specific models with regard to their different properties. These properties refer to the timing of the processes (here: time windows with a certain onset and duration), the reliability of classification within the person (BAC), and the topographical distribution of classification information (i.e., filters and patterns). For each property, I suggest an appropriate summary strategy that takes its corresponding characteristic into account. When, summarized in this way, the information can be taken to visualize the inter-individual variation and to compare the distribution of model properties across groups. Summarizing person-specific models on the group level poses a serious challenge both on a conceptual and on a methodological dimension. I regard my approach as a first step in this direction.

On the substantive level, in terms of classification accuracy, differences between the age groups were observed. With regard to the attentional focus prediction, the BAC was highest for older adults, followed by younger adults, and lowest for children. Concerning the WM load classification, the highest BACs were obtained for the younger adults, followed by the older adults, with the lowest values again for children.

In the group of younger adults, both prediction tasks revealed inter-individual differences in the onset and duration of time windows that discriminated optimally between conditions. In most younger adults, a model was selected that starts earlier than 250

ms after memory array onset. However, the fine mean BAC by time plot (see Fig. 5.6e) revealed two peaks in this group, one early peak at around 200 ms and a later one at around 700 ms. In older adults, I found a strong shift of the time windows towards stimulus onset. In all but one participant, the best model's onset was found to be before 200 ms. For a more detailed discussion of the substantive results, see Karch et al. (2015).

One limitation of my approach is that for the attentional focus prediction, the experimental condition remained the same for blocks of 30 trails each. If slowly varying background activity that could be exploited by the classification algorithm were present in a blocked experimental design, using k-fold cross-validation might overestimate the performance of the classifier. A general remedy would be to use leave-one-block-out cross-validation instead (Lemm et al., 2011).

An interesting topic for future work is to explore the amount of individualization necessary. While standard EEG analysis approaches assume the same model for each person, I made the assumption that a different model is required for every person. The two approaches represent the extremes of a continuum. It would be very interesting to systematically explore this continuum, that is, to examine how much individualization of the models is necessary. I envision two different approaches to reach this goal. The first would start with one model for all participants and recursively split the participants into groups with the same model. The participants are only split further if a significant gain in prediction accuracy can be obtained. Brandmaier et al. (2013) already developed this framework for SEMs. An alternative approach would be a hierarchical model that penalizes the person-specific models for a derivation from the group model. The best amount of penalty could be estimated and would be informative regarding the amount of individualization necessary for a given task. Both approaches are accompanied by their own methodological and computational challenges.

6.3. Conclusion

In sum, the present thesis introduces two novel modeling approaches for multiple individuals' repeated measures data based on machine learning methods. The first, GPPM aims at improving the modeling of panel data by, for example, extending the space of expressible models as compared to conventional panel methods. The second, a method to obtain person-specific models for EEG data, focuses on improving the appreciation of inter-individual variation in brain–behavior mappings in cognitive neuroscience, particularly in EEG data, by providing a way to obtain multivariate person-specific models. Both aspects, improving the analysis of panel data as well as developing modeling approaches that better account for idiosyncrasies in mappings between neural and cognitive processes, are necessary for progress in lifespan psychology.

References

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In E. Parzen, K. Tanabe, & G. Kitagawa (Eds.), *Selected papers of Hirotugu Akaike* (pp. 199–213). New York, NY: Springer. doi:10.1007/978-1-4612-1694-0_15

Ashburner, J. & Klöppel, S. (2011). Multivariate models of inter-subject anatomical variability. *NeuroImage*, *56*(2), 422–439. doi:10.1016/j.neuroimage.2010.03.059

Astle, D. E. & Scerif, G. (2011). Interactions between attention and visual short-term memory (VSTM): What can be learnt from individual and developmental differences? *Neuropsychologia*, *49*(6), 1435–1445. doi:10.1016/j.neuropsychologia.2010.12.001

Baltes, P. B. & Nesselroade, J. R. (1979). History and rationale of longitudinal research. In J. R. Nesselroade & P. B. Baltes (Eds.), *Longitudinal research in the study of behavior and development* (pp. 1–39). New York, NY: Academic Press.

Barrett, P. (2007). Structural equation modelling: Adjudging model fit. *Personality and Individual Differences*, *42*(5), 815–824. doi:10.1016/j.paid.2006.09.018

Berger, J. O. (1993). *Statistical decision theory and Bayesian analysis* (2nd ed.). New York, NY: Springer.

Bießmann, F., Daehne, S., Meinecke, F., Goergen, K., Blankertz, B., & Haufe, S. (2012). On the interpretability of linear multivariate neuroimaging analyses: Filters, patterns and their relationship. In *Proceedings of the 2nd NIPS Workshop on Machine Learning and Interpretation in Neuroimaging*. Retrieved from http://www.user.tu-berlin.de/felix.biessmann/pub/2012-MLINIFiltersandPatterns.pdf

Bishop, C. M. (2006). *Pattern recognition and machine learning* (1st ed.). New York, NY: Springer.

Blankertz, B., Lemm, S., Treder, M., Haufe, S., & Müller, K.-R. (2011). Single-trial analysis and classification of ERP components — A tutorial. *NeuroImage*, *56*, 814–825. doi:10.1016/j.neuroimage.2010.06.048

Blankertz, B., Tangermann, M., Vidaurre, C., Dickhaus, T., Sannelli, C., Popescu, F., ... Müller, K.-R. (2010). Detecting mental states by machine learning techniques: The Berlin brain–computer interface. In B. Graimann, G. Pfurtscheller, & B. Allison (Eds.), *Brain-computer interfaces* (pp. 113–135). Berlin: Springer.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Mülller, K.-R. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. doi:10.1109/MSP.2008.4408441

Boker, S. M. (2007a). Differential equation models for longitudinal data. In S. Menard (Ed.), *Handbook of longitudinal research* (pp. 639–652). New York, NY: Elsevier.

Boker, S. M. (2007b). Specifying latent differential equations models. In S. M. Boker & M. Wenger (Eds.), *Data analytic techniques for dynamical systems in the social and behavioral sciences*. Mahwah, NJ: Lawrence Erlbaum.

Bolger, N. & Laurenceau, J.-P. (2013). *Intensive longitudinal methods: An introduction to diary and experience sampling research* (1st ed.). New York, NY: Guilford.

Bollen, K. A. (1989). *Structural equations with latent variables* (1st ed.). New York, NY: Wiley.

Boyle, P. (2007). *Gaussian processes for regression and optimisation* (Doctoral Dissertation, Victoria University of Wellington). Retrieved from http://researcharchive.vuw.ac.nz/handle/10063/421

Brahim-Belhouari, S. & Bermak, A. (2004). Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, *47*(4), 705–712. doi:10.1016/j.csda.2004.02.006

Brandmaier, A. M., von Oertzen, T., McArdle, J. J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, *18*(1), 71–86. doi:10.1037/a0030001

Breakspear, M., Jirsa, V., & Deco, G. (2010). Computational models of the brain: From structure to function. *NeuroImage*, *52*(3), 727–730. doi:10.1016/j.neuroimage.2010.05.061

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, *16*(3), 199–231. Retrieved from http://projecteuclid.org/euclid.ss/1009213726

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (pp. 3121–3124). doi:10.1109/ICPR.2010.764

Burkholder, G. J. & Harlow, L. L. (2003). An illustration of a longitudinal cross-lagged design for larger structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *10*(3), 465–486. doi:10.1207/S15328007SEM1003_8

Burnham, K. P. (2013). *Model selection and multimodel inference: A practical information-theoretic approach* (2nd ed.). New York, NY: Springer.

Claeskens, G. & Hjort, N. L. (2008). *Model selection and model averaging*. Cambridge, UK: Cambridge University Press.

Cohen, J. (1968). Multiple regression as a general data-analytic system. *Psychological Bulletin*, *70*, 426–443. doi:10.1037/h0026714

Cohen, J., Cohen, J., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/-correlation analysis for the behavioral sciences* (3rd ed.). Machwach, NJ: Erlbaum.

Corbin, L. & Marquer, J. (2009). Individual differences in Sternberg's memory scanning task. *Acta Psychologica*, *131*(2), 153–162. doi:10.1016/j.actpsy.2009.04.001

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). Cambridge, MA: MIT Press.

Cox, G. E., Kachergis, G., & Shiffrin, R. M. (2012). Gaussian process regression for trajectory analysis. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society* (pp. 1440–1445). Retrieved from http://palm.mindmodeling.org/cogsci2012/papers/0256/paper0256.pdf

Csató, L. (2002). *Gaussian processes: Iterative sparse approximations* (Doctoral Dissertation, Aston University). Retrieved from http://eprints.aston.ac.uk/1327/

Cunningham, J., Ghahramani, Z., & Rasmussen, C. E. (2012). Gaussian processes for time-marked time-series data. In *International Conference on Artificial Intelligence and Statistics* (pp. 255–263). Retrieved from http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2012_CunninghamGR12.pdf

Curran, P. J. (2003). Have multilevel models been structural equation models all along? *Multivariate Behavioral Research*, *38*(4), 529–569. doi:10.1207/s15327906mbr3804_5

Damouras, S. (2008). *Nonparametric time series analysis using Gaussian processes* (Doctoral Dissertation, Carnegie Mellon University). Retrieved from http://www.utsc.utoronto.ca/~sdamouras/files/thesis.pdf

Danziger, K. (1990). *Constructing the subject: Historical origins of psychological research*. Cambridge, UK: Cambridge University Press.

Darwin, C. (1859). *On the origin of species*. London, England: John Murray.

Deco, G., Senden, M., & Jirsa, V. (2012). How anatomy shapes dynamics: A semi-analytical study of the brain at rest by a simple spin model. *Frontiers in Computational Neuroscience*, *6*(68). doi:10.3389/fncom.2012.00068

DeGroot, M. H. & Schervish, M. J. (2011). *Probability and statistics* (4th ed.). Boston, MA: Pearson.

Diaconis, P. (2009). The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, *46*(2), 179–205. doi:10.1090/S0273-0979-08-01238-X

Doppelmayr, M. M., Klimesch, W., Pachinger, T., & Ripper, B. (1998). The functional significance of absolute power with respect to event-related desynchronization. *Brain Topography*, *11*(2), 133–140. doi:10.1023/A:1022206622348

Doyle, O. M., Ashburner, J., Zelaya, F. O., Williams, S. C. R., Mehta, M. A., & Marquand, A. F. (2013). Multivariate decoding of brain images using ordinal regression. *NeuroImage*, *81*, 347–357. doi:10.1016/j.neuroimage.2013.05.036

Driver, C. C., Oud, J. H. L., & Voelkle, M. C. (in press). Continuous time structural equation modelling with R package ctsem. *Journal of Statistical Software*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York, NY: Wiley-Interscience.

Eid, M., Gollwitzer, M., & Schmitt, M. (2013). *Statistik und Forschungsmethoden [Statistics and research methods]* (1st ed.). Weinheim, Germany: Beltz.

Estabrook, R. & Neale, M. C. (2013). A comparison of factor score estimation methods in the presence of missing data: Reliability and an application to nicotine dependence. *Multivariate Behavioral Research*, *48*(1), 1–27. doi:10.1080/00273171.2012.730072

Fan, X. (1997). Canonical correlation analysis and structural equation modeling: What do they have in common? *Structural Equation Modeling*, *4*(1), 65–79. doi:10.1080/10705519709540060

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., & Grozea, C. (2009). Subject-independent mental state classification in single trials. *Neural Networks*, *22*(9), 1305–1312. doi:10.1016/j.neunet.2009.06.003

Fox, J., Nie, Z., & Byrnes, J. (2015). Sem: Structural equation models. Retrieved from http://CRAN.R-project.org/package=sem

Freunberger, R., Höller, Y., Griesmayr, B., Gruber, W., Sauseng, P., & Klimesch, W. (2008). Functional similarities between the p1 component and alpha oscillations. *European Journal of Neuroscience*, *27*(9), 2330–2340. doi:10.1111/j.1460-9568.2008.06190.x

Freytag, A., Rodner, E., Bodesheim, P., & Denzler, J. (2012). Beyond classification–large-scale Gaussian process inference and uncertainty prediction. In *Big Data Meets Computer Vision - NIPS 2012 Worksho*. Retrieved February 24, 2015, from http://www.inf-cv.uni-jena.de/dbvmedia/de/Research/GP_HIK/Freytag12_BCL.pdf

Friedman, J. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, *84*(405), 165–175.

Friston, K., Chu, C., Mourão-Miranda, J., Hulme, O., Rees, G., Penny, W., & Ashburner, J. (2008). Bayesian decoding of brain images. *NeuroImage*, *39*(1), 181–205. doi:10.1016/j.neuroimage.2007.08.013

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). Boca Raton, FL: CRC.

Griffiths, T. L., Lucas, C., Williams, J., & Kalish, M. L. (2009). Modeling human function learning with Gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 553–560). Retrieved from http://papers.nips.cc/paper/3529-modeling-human-function-learning-with-gaussian-processes

Grimault, S., Robitaille, N., Grova, C., Lina, J.-M., Dubarry, A.-S., & Jolicœur, P. (2009). Oscillatory activity in parietal and dorsolateral prefrontal cortex during retention in visual short-term memory: Additive effects of spatial attention and memory load. *Human Brain Mapping*, *30*(10), 3378–3392. doi:10.1002/hbm.20759

Hall, P., Müller, H.-G., & Yao, F. (2008). Modelling sparse generalized longitudinal observations with latent Gaussian processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(4), 703–723. doi:10.1111/j.1467-9868.2008.00656.x

Hartikainen, J. & Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 379–384). doi:10.1109/MLSP.2010.5589113

Hastie, T., Tibshirani, R., & Friedman, J. J. H. (2001). *The elements of statistical learning*. New York, NY: Springer.

Heitmeyer, W. (Ed.). (2004). *Deutsche Zustände. Folge 3 [Current state in Germany. Series 3]*. Frankfurt am Main: Suhrkamp.

Hertzog, C. & Nesselroade, J. R. (2003). Assessing psychological change in adulthood: An overview of methodological issues. *Psychology and Aging*, *18*(4), 639–657. doi:10.1037/0882-7974.18.4.639

Hoyle, R. H. (Ed.). (2014, November). *Handbook of structural equation modeling*. New York, NY: Guilford.

Hughes, N. J., Hunt, J. J., Cloherty, S. L., Ibbotson, M. R., Sengpiel, F., & Goodhill, G. J. (2014). Stripe-rearing changes multiple aspects of the structure of primary visual cortex. *NeuroImage*, *95*, 305–319. doi:10.1016/j.neuroimage.2014.03.031

Jordan, M. I. (2009). *Bayesian or frequentist, which are you? [Recorded Lecture]*. Cambridge, UK. Retrieved March 18, 2015, from http://videolectures.net/mlss09uk_jordan_bfway/

Jung, T., Makeig, S., Humphries, C., Lee, T., McKeown, M., Iragui, V., & Sejnowski, T. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, *37*, 163–178. doi:10.1111/1469-8986.3720163

Kaden, E., Anwander, A., & Knösche, T. R. (2008). Variational inference of the fiber orientation density using diffusion MR imaging. *NeuroImage*, *42*(4), 1366–1380. doi:10.1016/j.neuroimage.2008.06.004

Kaplan, D. (2009). *Structural equation modeling: Foundations and extensions* (2nd ed.). Thousand Oaks, CA: Sage.

Karch, J. D., Sander, M. C., von Oertzen, T., Brandmaier, A. M., & Werkle-Bergner, M. (2015). Using within-subject pattern classification to understand lifespan age differences in oscillatory mechanisms of working memory selection and maintenance. *NeuroImage*, *118*, 538–552. doi:10.1016/j.neuroimage.2015.04.038

Kauppi, J.-P., Kandemir, M., Saarinen, V.-M., Hirvenkari, L., Parkkonen, L., Klami, A., ... Kaskimy, S. (2015). Towards brain-activity-controlled information retrieval: Decoding image relevance from MEG signals. *NeuroImage*. doi:10.1016/j.neuroimage.2014.12.079

Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning*, *27*(1), 7–50. doi:10.1023/A:1007344726582

Kelly, S. P., Lalor, E. C., Reilly, R. B., & Foxe, J. J. (2006). Increases in alpha oscillatory power reflect an active retinotopic mechanism for distracter suppression during sustained visuospatial attention. *Journal of Neurophysiology*, *95*(6), 3844–3851. doi:10.1152/jn.01234.2005

Klimesch, W. (1999). EEG alpha and theta oscillations reflect cognitive and memory performance: A review and analysis. *Brain Research Reviews*, *29*(2–3), 169–195. doi:10.1016/S0165-0173(98)00056-3

Kline, R. B. (2011). *Principles and practice of structural equation modeling* (3rd ed.). New York, NY: Guilford.

Knapp, T. R. (1978). Canonical correlation analysis: A general parametric significance-testing system. *Psychological Bulletin*, *85*(2), 410–416. doi:10.1037/0033-2909.85.2.410

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence*. Retrieved June 20, 2016, from http://robotics.stanford.edu/~ronnyk/accEst.pdf

Kostro, D., Abdulkadir, A., Durr, A., Roos, R., Leavitt, B. R., Johnson, H., ... Klöppel, S. (2014). Correction of inter-scanner and within-subject variance in structural

MRI based automated diagnosing. *NeuroImage*, *98*, 405–415. doi:10.1016/j.neuroimage.2014.04.057

Kraft, D. (1994). Algorithm 733: TOMP–Fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, *20*(3), 262–281. doi:10.1145/192115.192124

Kray, J. & Lindenberger, U. (2000). Adult age differences in task switching. *Psychology and Aging*, *15*(1), 126–147. doi:10.1037/0882-7974.15.1.126

Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S., & Baker, C. I. (2009). Circular analysis in systems neuroscience: The dangers of double dipping. *Nature Neuroscience*, *12*(5), 535–540. doi:10.1038/nn.2303

Kuo, H.-H. (2006). *Introduction to stochastic integration*. New York, NY: Springer.

Ledoit, O. & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 365–411. doi:10.1016/S0047-259X(03)00096-4

Lee, M. D. & Wagenmakers, E.-J. (2005). Bayesian statistical inference in psychology: Comment on Trafimow (2003). *Psychological Review*, *112*(3), 662–668. doi:10.1037/0033-295X.112.3.662

Leibniz, G. W. (1704/1886). *Nouveaux essais sur l'entendement humain [New essays on human understanding]*. Paris, France: Hachette.

Lemm, S., Blankertz, B., Dickhaus, T., & Müller, K.-R. (2011). Introduction to machine learning for brain imaging. *NeuroImage*, *56*(2), 387–399. doi:10.1016/j.neuroimage.2010.11.004

Lindenberger, U., Burzynska, A., & Nagel, I. E. (2013). Heterogeneity in frontal-lobe aging. In D. T. Stuss & R. T. Knight (Eds.), *Principles of frontal lobe function* (Vol. 2, pp. 609–627). New York, NY: Oxford University Press. doi:10.1093/med/9780199837755.003.0043

Little, T. D. (2013). *Longitudinal structural equation modeling*. New York, NY: Guilford.

Liu, Z., Wu, L., & Hauskrecht, M. (2013). Modeling clinical time series using Gaussian process sequences. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (pp. 623–631). doi:10.1137/1.9781611972832.69

Luck, S. J. (2005). *An introduction to the event-related potential technique*. Cambridge, MA: MIT Press.

Luck, S. J. & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*(6657), 279–281. doi:10.1038/36846

Macke, J. H., Gerwinn, S., White, L. E., Kaschube, M., & Bethge, M. (2011). Gaussian process methods for estimating cortical maps. *NeuroImage*, *56*(2), 570–581. doi:10.1016/j.neuroimage.2010.04.272

Markowetz, A., Błaszkiewicz, K., Montag, C., Switala, C., & Schlaepfer, T. E. (2014). Psycho-informatics: Big Data shaping modern psychometrics. *Medical Hypotheses*, *82*(4), 405–411. doi:10.1016/j.mehy.2013.11.030

Marquand, A. F., Brammer, M., Williams, S. C. R., & Doyle, O. M. (2014). Bayesian multi-task learning for decoding multi-subject neuroimaging data. *NeuroImage*, *92*, 298–311. doi:10.1016/j.neuroimage.2014.02.008

Marquand, A. F., Howard, M., Brammer, M., Chu, C., Coen, S., & Mourão-Miranda, J. (2010). Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. *NeuroImage*, *49*(3), 2178–2189. doi:10.1016/j.neuroimage.2009.10.072

McArdle, J. J. & McDonald, R. P. (1984). Some algebraic properties of the Reticular Action Model for moment structures. *British Journal of Mathematical and Statistical Psychology*, *37*(2), 234–251. doi:10.1111/j.2044-8317.1984.tb00802.x

Molenaar, P. C. M. (2013). On the necessity to use person-specific data analysis approaches in psychology. *European Journal of Developmental Psychology*, *10*(1), 29–39. doi:10.1080/17405629.2012.747435

Molenaar, P. C. M. & Campbell, C. G. (2009, April 1). The new person-specific paradigm in psychology. *Current Directions in Psychological Science*, *18*(2), 112–117. doi:10.1111/j.1467-8721.2009.01619.x

Müller-Gerking, J., Pfurtscheller, G., & Flyvbjerg, H. (1999). Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, *110*(5), 787–798.

Murphy, K. P. (2012, August). *Machine learning: A probabilistic perspective*. Cambridge, MA: MIT Press.

Muthén, L. & Muthén, B. (1998–2012). *Mplus user's guide* (7th ed.). Los Angeles, CA: Muthén & Muthén.

Nagel, I. E., Preuschhof, C., Li, S.-C., Nyberg, L., Bäckman, L., Lindenberger, U., & Heekeren, H. R. (2009). Performance level modulates adult age differences in brain activation during spatial working memory. *Proceedings of the National Academy of Sciences*, *106*(52), 22552–22557. doi:10.1073/pnas.0908238106

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., ... Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, *81*, 535–549. doi:10.1007/s11336-014-9435-8

Nesselroade, J. R., Gerstorf, D., Hardy, S. A., & Ram, N. (2007). Focus article: Idiographic filters for psychological constructs. *Measurement*, *5*(4), 217–235. doi:10.1080/15366360701741807

Nettleton, D. & Doerge, R. W. (2000). Accounting for variability in the use of permutation testing to detect quantitative trait loci. *Biometrics*, *56*(1), 52–58. Retrieved from http://www.jstor.org/stable/2677102

Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J.-M. (2010). Fieldtrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, *2011*. doi:10.1155/2011/156869

Oud, J. H. L. & Jansen, R. A. R. G. (2000). Continuous time state space modeling of panel data by means of sem. *Psychometrika*, *65*(2), 199–215. doi:10.1007/BF02294374

Oud, J. H. L. & Singer, H. (2008). Continuous time modeling of panel data: SEM versus filter techniques. *Statistica Neerlandica*, *62*(1), 4–28. doi:10.1111/j.1467-9574.2007.00376.x

Park, S. & Choi, S. (2010). Hierarchical Gaussian process regression. In *2nd Asian Conference on Machine Learning* (pp. 95–110). Retrieved from http://www.jmlr.org/proceedings/papers/v13/park10a/park10a.pdf

Parra, L. C., Spence, C. D., Gerson, A. D., & Sajda, P. (2005). Recipes for the linear analysis of EEG. *NeuroImage*, *28*, 326–341. doi:10.1016/j.neuroimage.2005.05.032

Pek, J. & Wu, H. (2015). Profile likelihood-based confidence intervals and regions for structural equation models. *Psychometrika*, *80*(4), 1123–1145. doi:10.1007/s11336-015-9461-1

Pfurtscheller, G. & Aranibar, A. (1977). Event-related cortical desynchronization detected by power measurements of scalp EEG. *Electroencephalography and Clinical Neurophysiology*, *42*(6), 817–826. doi:10.1016/0013-4694(77)90235-8

Philiastides, M. G. & Sajda, P. (2006). Temporal characterization of the neural correlates of perceptual decision making in the human brain. *Cerebral Cortex*, *16*(4), 509–518. doi:10.1093/cercor/bhi130

Prokhorov, Y. (n.d.). *Random variable*. In *Encyclopedia of mathematics*. Retrieved from http://www.encyclopediaofmath.org/index.php?title=Random_variable&oldid=29510

R Core Team. (2015). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

Ramoser, H., Müller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, *8*(4), 441–446. doi:10.1109/86.895946

Rasmussen, C. E. (2006). Gaussian processes for machine learning. Retrieved April 21, 2015, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.3414

Rasmussen, C. E. & Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, *11*, 3011–3015. Retrieved from http://www.jmlr.org/papers/volume11/rasmussen10a/rasmussen10a.pdf

Rasmussen, C. E. & Nickisch, H. (2015). Documentation for GPML Matlab code version 3.6. Retrieved June 15, 2016, from http://www.gaussianprocess.org/gpml/code/matlab/doc/

Rice, J. A. (2007). *Mathematical statistics and data analysis* (3rd ed.). Belmont, CA: Thomson Higher Education.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., & Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A*, *371*(1984), 20110550. doi:10.1098/rsta.2011.0550. PMID: 23277607

Rosseel, Y. (2012). Lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. doi:10.18637/jss.v048.i02

Royston, P. (1995). Algorithm as r94. *Applied Statistics*, *44*(4).

Ruigrok, A. N. V., Salimi-Khorshidi, G., Lai, M.-C., Baron-Cohen, S., Lombardo, M. V., Tait, R. J., & Suckling, J. (2014). A meta-analysis of sex differences in human brain structure. *Neuroscience and Biobehavioral Reviews*, *39*, 34–50. doi:10.1016/j.neubiorev.2013.12.004

Saatçi, Y., Turner, R. D., & Rasmussen, C. E. (2010). Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 927–934). Retrieved July 8, 2015, from http://machinelearning.wustl.edu/mlpapers/paper_files/icml2010_SaatciTR10.pdf

Salimi-Khorshidi, G., Nichols, T., Smith, S., & Woolrich, M. (2011). Using Gaussian-process regression for meta-analytic neuroimaging inference based on sparse observations. *IEEE Transactions on Medical Imaging*, *30*(7), 1401–1416. doi:10.1109/TMI.2011.2122341

Saltelli, A., Chan, K., & Scott, E. (2000). *Sensitivity analysis* (1st ed.). Chichester, England: Wiley.

Sander, M. C., Werkle-Bergner, M., & Lindenberger, U. (2012). Amplitude modulations and inter-trial phase stability of alpha-oscillations differentially reflect working memory constraints across the lifespan. *NeuroImage*, *59*, 646–654. doi:10.18637/jss.v048.i02

Särkkä, S. & Hartikainen, J. (2012). Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In *International Conference on Artificial Intelligence and Statistics* (pp. 993–1001). Retrieved May 24, 2016, from %5C%7Bhttp://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2012_SarkkaH12.pdf%5C%7D

SAS Institute Inc. (2015). SAS 9.4 [Computer software]. Retrieved October 16, 2015, from http://www.sas.com/en_us/software/sas9.html

Sauseng, P., Klimesch, W., Heise, K., Gruber, W., Holz, E., Karim, A., ... Hummel, F. (2009). Brain oscillatory substrates of visual short-term memory capacity. *Current Biology*, *19*, 1846–1852. doi:10.1016/j.cub.2009.08.062

Schmiedek, F., Bauer, C., Lövdén, M., Brose, A., & Lindenberger, U. (2010). Cognitive enrichment in old age: Web-based training programs. *GeroPsych*, *23*(2), 59–67. doi:10.1024/1662-9647/a000013

Scientific Software International Inc. (2015). LISREL 9 [Computer software]. Retrieved October 16, 2015, from http://www.ssicentral.com/lisrel/

Sivo, S., Fan, X., & Witta, L. (2005). The biasing effects of unmodeled ARMA time series processes on latent growth curve model estimates. *Structural Equation Modeling: A Multidisciplinary Journal*, *12*(2), 215–231. doi:10.1207/s15328007sem1202_2

Statsoft Inc. (2015). STATISTICA [Computer Software]. Retrieved October 16, 2015, from http://www.statsoft.com/Products/STATISTICA/Product-Index

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 111–147.

Taboga, M. (2012a). Convergence in distribution. In *Lectures on Probability Theory and Mathematical Statistics*. Lexington, KY: CreateSpace Independent Publishing Platform. Retrieved from http://www.statlect.com/subon2/dsconv1.htm

Taboga, M. (2012b). *Lectures on probability theory and mathematical statistics* (2nd ed.). Lexington, KY: CreateSpace Independent Publishing Platform.

Taboga, M. (2012c). Likelihood ratio test. In *Lectures on Probability Theory and Mathematical Statistics* (2nd ed.). Lexington, KY: CreateSpace Independent Publishing Platform. Retrieved from http://www.statlect.com/likelihood_ratio_test.htm

Taboga, M. (2012d). Maximum likelihood. In *Lectures on Probability Theory and Mathematical Statistics* (2nd ed.). Lexington, KY: CreateSpace Independent Publishing Platform. Retrieved March 29, 2016, from http://www.statlect.com/fundamentals-of-statistics/maximum-likelihood

Tomarken, A. J. & Waller, N. G. (2003). Potential problems with "well fitting" models. *Journal of Abnormal Psychology*, *112*(4), 578–598. doi:10.1037/0021-843X.112.4.578

Turner, R. D. (2012). *Gaussian processes for state space models and change point detection* (Doctoral Dissertation, University of Cambridge). Retrieved July 8, 2015, from https://www.repository.cam.ac.uk/handle/1810/242181

Vaden, R. J., Hutcheson, N. L., McCollum, L. A., Kentros, J., & Visscher, K. M. (2012). Older adults, unlike younger adults, do not modulate alpha power to suppress irrelevant information. *NeuroImage*, *63*(3), 1127–1133. doi:10.1016/j.neuroimage.2012.07.050

Valdes-Hernandez, P. A., Ojeda-Gonzalez, A., Martinez-Montes, E., Lage-Castellanos, A., Virues-Alba, T., Valdes-Urrutia, L., & Valdes-Sosa, P. A. (2010). White matter architecture rather than cortical surface area correlates with the EEG alpha rhythm. *NeuroImage*, *49*(3), 2328–2339. doi:10.1016/j.neuroimage.2009.10.030

Varma, S. & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(1), 91. doi:10.1186/1471-2105-7-91

Voelkle, M. C. (in press). A new perspective on three old methodological issues: The role of time, missing values, and cohorts in longitudinal models of youth development. In A. C. Petersen, S. H. Koller, S. Verma, & F. Motti-Stefanidi (Eds.), *Positive youth development in global contexts of social and economic change*. Hove, England: Psychology Press.

Voelkle, M. C., Oud, J. H., Davidov, E., & Schmidt, P. (2012). An SEM approach to continuous time modeling of panel data: Relating authoritarianism and anomia. *Psychological Methods*, *17*(2), 176–192. doi:10.1037/a0027543

Vogel, E. & Machizawa, M. (2004). Neural activity predicts individual differences in visual working memory capacity. *Nature*, *428*, 748–751. doi:10.1038/nature02447

von Oertzen, T., Brandmaier, A. M., & Tsang, S. (2015). Structural equation modeling with Ωnyx. *Structural Equation Modeling*, *22*(1), 148–161. doi:10.1080/10705511.2014.935842

von Oertzen, T. & Brick, T. R. (2014). Efficient Hessian computation using sparse matrix derivatives in RAM notation. *Behavior Research Methods*, *46*, 385–395. doi:10.3758/s13428-013-0384-4

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., & van der Maas, H. L. J. (2011). Why psychologists must change the way they analyze their data: The case of psi: Comment on Bem (2011). *Journal of Personality and Social Psychology*, *100*(3), 426–432. doi:10.1037/a0022790

Wasserman, L. (2004). *All of statistics: A concise course in statistical inference*. New York, NY: Springer.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, *54*(6), 1063–1070. doi:10.1037/0022-3514.54.6.1063

Werkle-Bergner, M., Freunberger, R., Sander, M. C., Lindenberger, U., & Klimesch, W. (2012). Inter-individual performance differences in younger and older adults differentially relate to amplitude modulations and phase stability of oscillations controlling working memory contents. *NeuroImage*, *60*(1), 71–82. doi:10.1016/j.neuroimage.2011.11.071

Wolpaw, J. R. & Wolpaw, E. W. (2012). *Brain-computer interfaces: Principles and practice* (1st ed.). Oxford University Press.

Worden, M. S., Foxe, J. J., Wang, N., & Simpson, G. V. (2000). Anticipatory biasing of visuospatial attention indexed by retinotopically specific alpha-band electroencephalography increases over occipital cortex. *The Journal of Neuroscience*, *20*(6), RC63. Retrieved from http://www.jneurosci.org/content/20/6/RC63.long

Yarkoni, T. (2012). Psychoinformatics: New horizons at the interface of the psychological and computing sciences. *Current Directions in Psychological Science*, *21*(6), 391–397. doi:10.1177/0963721412457362

Ziegler, G., Ridgway, G. R., Dahnke, R., & Gaser, C. (2014). Individualized Gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. *NeuroImage*, *97*, 333–348. doi:10.1016/j.neuroimage.2014.04.018

A. Probability Theory

A.1. Foundations of Probability Theory

Definition A.1.1. A *probability space* is a triple $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is a sample space, $\mathcal{F}$ a σ-algebra on the power set of the sample space 2^{Ω}, and $\mathbb{P}$ a probability measure. The sample space Ω is an arbitrary non-empty set.
A *σ-algebra* $\mathcal{F}$ must satisfy the following properties:

1. $\Omega \in \mathcal{F}$
2. $A \in \mathcal{F} \Rightarrow A^{\complement} \in \mathcal{F}$, where $A^{\complement} = \Omega \setminus A$
3. If $A_n \in \mathcal{F}$ for all $n \in \mathbb{N}$, it follows that $\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{F}$

The elements $A \in \mathcal{F}$ are called *events*. The elements $\omega \in \Omega$ are called outcomes. A function $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ is a *probability measure* on the sample space σ-algebra pair $(\Omega, \mathcal{F})$ if it satisfies the following properties:

1. $\mathbb{P}(A) \geq 0$ for every $A \in \mathcal{F}$
2. $\mathbb{P}(\Omega)$=1
3. If $A_n \in \mathcal{F}$ for all $n \in \mathbb{N}$ and for any two events A_n, A_m for which $n \neq m$, $A_n \cap A_m = \emptyset$, it follows that $\mathbb{P}(\bigcup_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} \mathbb{P}(A_n)$

Definition A.1.2. A *random variable* on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a mapping

$$X : \Omega \to \mathbb{R}$$

that assigns a real number $X(\omega)$ to each outcome $\omega \in \Omega$ and fulfills the following condition (Prokhorov, n.d.):

$$\text{for all } x \in \mathbb{R} : \{\omega : X(\omega) \leq x\} \in \mathcal{F}.$$

Definition A.1.3. A *random vector* X is a n-dimensional vector $X = [X_1, \ldots, X_n]$ of random variables X_i. All X_i are random variables on the same probability space.

Remark A.1.4. A random variable is a 1-dimensional random vector.

Theorem A.1.5. Let X be a n-dimensional random vector on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $\mathcal{F}_X$ the class of subsets $\mathcal{F}_X \subset 2^{\mathbb{R}^n}$ with corresponding members $A_X \in \mathcal{F}_X$ for which

$$\{\omega : [X_1(\omega), \ldots, X_n(\omega)] \in \mathcal{A}_X\} \in \mathcal{F},$$

then $\mathcal{F}_X$ is a σ-algebra. The function $\mathbb{P}_X$ defined on $\mathcal{F}_X$ by

$$\mathbb{P}_X(A_X) = \mathbb{P}(\{\omega : [X_1(\omega), \ldots, X_n(\omega)] \in A_X\})$$

is a probability measure. Hence, the triple $(\mathbb{R}^n, \mathcal{F}_X, \mathbb{P}_X)$ is a probability space. It is called the probability space induced by the random variable X. For every outcome $\omega \in \Omega$ the vector $X(w)$ is called a realization. The sample space $\mathbb{R}^n$ is called the support of the random vector X. (generalization of the treatment in Prokhorov [n.d.])

Remark A.1.6. I denote the set $\{\omega : [X_1(\omega), \ldots, X_n(\omega)] \in A_X\}$ as simply $\{X \in A_X\}$ in the remainder.

Definition A.1.7. Let $(\mathbb{R}^n, \mathcal{F}_X, \mathbb{P}_X)$ be a probability space induced by a random vector X. A random vector X is *continuous* iff a function p_X exists such that $p_X(x) \geq 0$ for all realizations $x = X(w)$, $\int_{\mathbb{R}^n} p_X(x)\mathrm{d}x = 1$, and for every event $A_X \in \mathcal{F}_X$

$$\mathbb{P}_X(A_X) = \int_{A_X} p_X(x)\mathrm{d}x.$$

The function p_X is called the (joint) *probability density function (pdf)* of X. (generalization of Wasserman, 2004, p. 23)

Remark A.1.8. There are also discrete random variables. The equivalent function for them is the probability mass function. In this work only continuous random vectors are considered. Therefore, the rest of the treatment will be limited to pdfs. However, note that everything can easily be generalized to discrete random variables. The difference is more or less that all integrals are exchanged by sums.

Definition A.1.9. The *expected value* of a continuous random vector $X : \Omega \to \mathbb{R}^n$ with pdf p_X is defined as

$$\mathbb{E}(X) = \int_{\mathbb{R}^n} p_X(x)x\mathrm{d}x.$$

Theorem A.1.10. Let $X : \Omega \to \mathbb{R}^n$ be a random vector and $r : \mathbb{R}^n \to \mathbb{R}^m$ a function, then $Y = r(X)$ is also a random vector.

A.2. Conditional Distributions and Independence

Definition A.2.1. A set of events $\{A_i : i \in I\}$ is *mutually independent* iff

$$\mathbb{P}\left(\bigcap_{i \in J} A_i\right) = \prod_{i \in J} \mathbb{P}(A_i)$$

for every finite subset J of I. (Wasserman, 2004, p. 25)

Definition A.2.2. Let A, B be two events. If $\mathbb{P}(B) > 0$, then the conditional probability of A given B is

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Bayes' rule follows From the definition of conditional probability.

Theorem A.2.3. Let A, B be two events and $\mathbb{P}(B) > 0$, then

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}.$$

The notation of conditional probability and independence can be extended to random vectors.

Definition A.2.4. A set of random vectors $\{X_i : i \in I\}$ is mutually independent iff

$$\mathbb{P}\left(\bigcap_{i \in J} \{X_i \in A_i\}\right) = \prod_{i \in J} \mathbb{P}(\{X_i \in A_i\})$$

for every finite subset J of I. This has to hold no matter which event $A_i \in A_{X_i}$ is chosen for a random vector X_i.

For continuous random vectors mutual independence can also be expressed in the form of pdfs.

Theorem A.2.5. n continuous random vectors X_i with corresponding pdfs $p_{X_i}(x_i)$ are mutually independent iff the joint pdf p_X can be decomposed as follows:

$$p_X(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_{X_i}(x_i).$$

Thus, if mutual dependence holds, the joint pdf of n random vectors can easily be obtained given the pdf of each random vector X_i. The other direction is also often important: Obtaining the pdf of one random vector Y given a joint pdf that involves Y.

Theorem A.2.6. Let $p_{X,Y}(x, y)$ be the joint pdf of the random vectors X and Y, then the marginal pdf of the random vector Y is obtained by

$$\int_{\mathbb{R}^n} p_{XY}(x, y)\mathrm{d}x,$$

where n is the dimensionality of the random vector X.

Remark A.2.7. Concatenating random vectors results in a random vector. Therefore, this theorem generalizes to all possible scenarios. X is simply defined as the random vector containing all the random vectors for which the marginal distribution is desired and Y as the remaining random vectors.

Let X, Y be two random vectors. Note that $\{X \in A_X\}$ and $\{Y \in A_Y\}$ are events. Therefore, $\mathbb{P}(\{X \in A_X\}|\{Y \in A_Y\})$ is a conditional probability. Furthermore, for a fixed event B $\mathbb{P}(\cdot|B)$ is a probability measure. Often of special interest is the conditioning on one realization of Y, i.e. $|A_Y| = 1$. These observations lead to the conditional probability distribution, which is expressed as a conditional pdf for continuous random variables.

Definition A.2.8. Let X, Y be two random vectors, then a function $p_{X|Y=y}$ is the conditional pdf of X given $Y = y$ iff

$$\mathbb{P}(X \in A_X|Y = y) = \int_{A_X} p_{X|Y=y}(x)\mathrm{d}x.$$

The question now of course is how to obtain the conditional pdf. All that is needed is the joint pdf of X and Y.

Theorem A.2.9. Let $[X, Y]$ be a partitioned continuous random vector, $p_{XY}(x, y)$ its joint pdf, and $p_Y(y)$ the marginal pdf of Y, then the conditional pdf of X given $Y = y$

is given by

$$p_{X|Y=y}(x) = \frac{p_{XY}(x,y)}{p_Y(y)}.$$

This fact leads to the most important rule for Bayesian inference, Bayes' rule for pdfs.

Theorem A.2.10.

$$p_{X|Y=y}(x) = \frac{p_{Y|X=x}(x)p_X(x)}{p_Y(y)}$$

Using marginalization, one obtains

$$p_{X|Y=y}(x) = \frac{p_{Y|X=x}(x)p_X(x)}{\int_{\Omega_Y} p_{XY}(x,y)\mathrm{d}y}.$$

Using Theorem A.2.9, the joint pdf can be factorized into the marginal and the conditional pdf:

$$p_{X|Y=y}(x) = \frac{p_{Y|X=x}(x)p_X(x)}{\int_{\Omega_Y} p_{X|Y=y}(x)p_Y(y)\mathrm{d}y}.$$

Remark A.2.11. The conditional pdf is often written as $p_{X|Y}(x|y)$. This notation is a little sloppy, as $p_{X|Y}(x|y)$ is not a pdf. In the applied literature and especially in the Bayesian literature, the notation is even simplified one step further to $p(x|y)$. Indeed, p is used a placeholder for any pdf and the respective pdf is identified only by its arguments. While in a purely mathematical sense this is wrong, I will use this notation in this work as it is so widespread.

A.3. (Co)Variance Rules and the Gaussian Distribution

Definition A.3.1. Let X, Y be two random variables, then

$$\mathrm{Cov}(X,Y) = \mathbb{E}([X - \mathbb{E}(X)][Y - \mathbb{E}(Y)])$$

is called *covariance* between X and Y. The special case $\mathrm{Cov}(X,X)$ is called *variance*.

Furthermore, for a random vector $X : \Omega \to \mathbb{R}^n$ the matrix

$$\begin{aligned}\mathrm{Cov}(X) &= \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & & \vdots \\ \vdots & & \ddots & \\ \mathrm{Cov}(X_n, X_1) & \dots & & \mathrm{Cov}(X_n, X_n) \end{bmatrix} \\ &= \mathbb{E}\left([X - \mathbb{E}(X)][X - \mathbb{E}(X)]^\top\right)\end{aligned}$$

is called the covariance matrix.

Theorem A.3.2. Let $A \in \mathbb{R}^{m \times n}$ be a matrix, $b \in \mathbb{R}^m$ a vector, and X a n-dimensional random vector with expected value μ and covariance matrix Σ, then the random vector $Z = AX + b$ has the expected value and covariance matrix

$$\mathbb{E}(Z) = A\mu + b, \mathrm{Cov}(Z) = A\Sigma A^\top.$$

Proof:

$$\begin{aligned}\mathbb{E}(AX + b) &= \int_{\mathbb{R}^n} p(x)(Ax + b)\mathrm{d}x = \int_{\mathbb{R}^n} p(x)Ax\mathrm{d}x + \int_{\mathbb{R}^n} p(x)b\mathrm{d}x \\ &= A\int_{\mathbb{R}^n} p(x)x\mathrm{d}x + b\int_{\mathbb{R}^n} p(x)\mathrm{d}x = A\mu + b \\ \mathrm{Cov}(AX + b) &= \mathbb{E}\left([AX + b - \mathbb{E}(AX + b)][AX + b - \mathbb{E}(AX + b)]^\top\right) \\ &= \mathbb{E}\left([AX + b - A\mathbb{E}(X) - b][AX + b - A\mathbb{E}(X) - b]^\top\right) \\ &= \mathbb{E}\left(A[X - \mathbb{E}(X)][X - \mathbb{E}(X))^\top]A^\top\right) \\ &= A\mathbb{E}\left([X - \mathbb{E}(X)][X - \mathbb{E}(X)]^\top\right)A^\top \\ &= A\Sigma A^\top\end{aligned}$$

□

The central probability distribution for this work is the multivariate Gaussian distribution.

Definition A.3.3. A random vector $X : \Omega \to \mathbb{R}^k$ has a multivariate Gaussian probability distribution iff its pdf is of the form

$$p(x) = \pi^{\frac{k}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(x - \mu)^\top\Sigma^{-1}(x - \mu)\right).$$

The parameters of the pdf are μ and Σ. $X \sim \mathcal{N}(\mu, \Sigma)$ denotes that the probability

distribution of the random vector X is the multivariate Gaussian distribution, with parameters μ and Σ.

Remark A.3.4. In this work the adjective Gaussian is used. If a random vector is Gaussian, it means that the probability distribution of the random vector is the multivariate Gaussian distribution.

Theorem A.3.5. The expected value of a Gaussian random vector $X \sim N(\mu, \Sigma)$ is $\mathbb{E}(X) = \mu$, and the covariance matrix is $\mathrm{Cov}(X) = \Sigma$. Thus, the probability distribution of a Gaussian random vector is fully described by its expected value and covariance matrix.

Remark A.3.6. Therefore, the parameters of the Gaussian distribution are referred to as the mean vector and covariance matrix. Mean is simply another word for expected value.

Theorem A.3.7. If the random vector X is Gaussian, then the random vector $AX + b$ is also Gaussian.

Corollary A.3.8. Let A be a matrix, b a vector, and X a Gaussian random vector with distribution $X \sim \mathcal{N}(\mu, \Sigma)$, then the random vector $Y = AX + b$ is also Gaussian with distribution $Y \sim \mathcal{N}(A\mu + b, A\Sigma A^\top)$.

Theorem A.3.9. Let $Z = [X, Y]^\top$ be a partition of a Gaussian random vector Z such that

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{XY}^\top & \Sigma_{YY} \end{bmatrix} \right),$$

then

1. The marginal distribution of X is $X \sim N(\mu_X, \Sigma_X)$.
2. The conditional distribution of Y given $X = x$ is

$$Y|X = x \sim \mathcal{N}(\mu_Y + \Sigma_{XY}^\top \Sigma_{XX}(x - \mu_x), \Sigma_{XX} - \Sigma_{XY}^\top \Sigma_{XX}^{-1} \Sigma_{XY}).$$

Theorem A.3.10. Let $X_1, \ldots, X_n$ be n mutually independent Gaussian random vectors

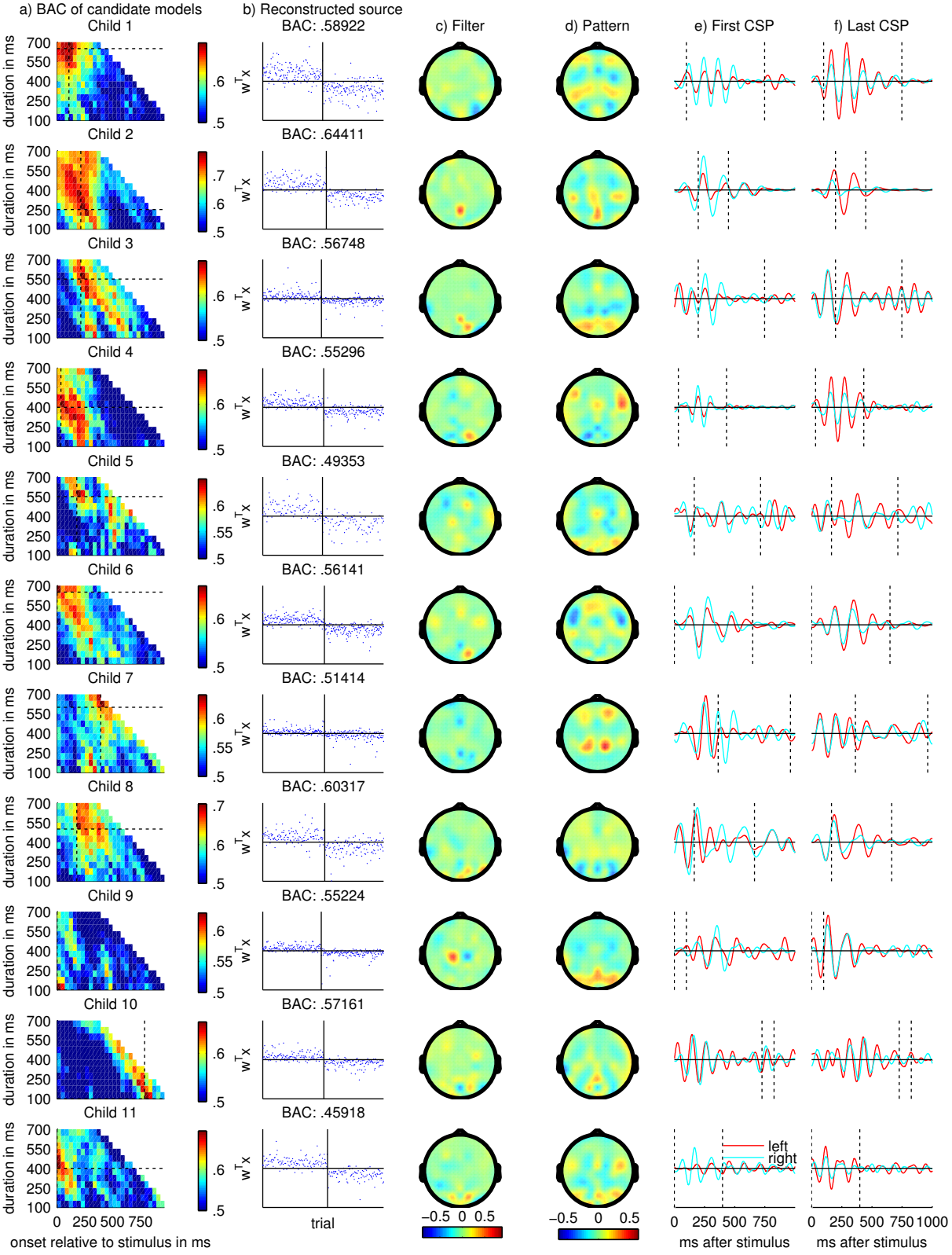
with $X_i \sim \mathcal{N}(\mu_i, \Sigma_i)$, then the joint distribution of these random vectors is

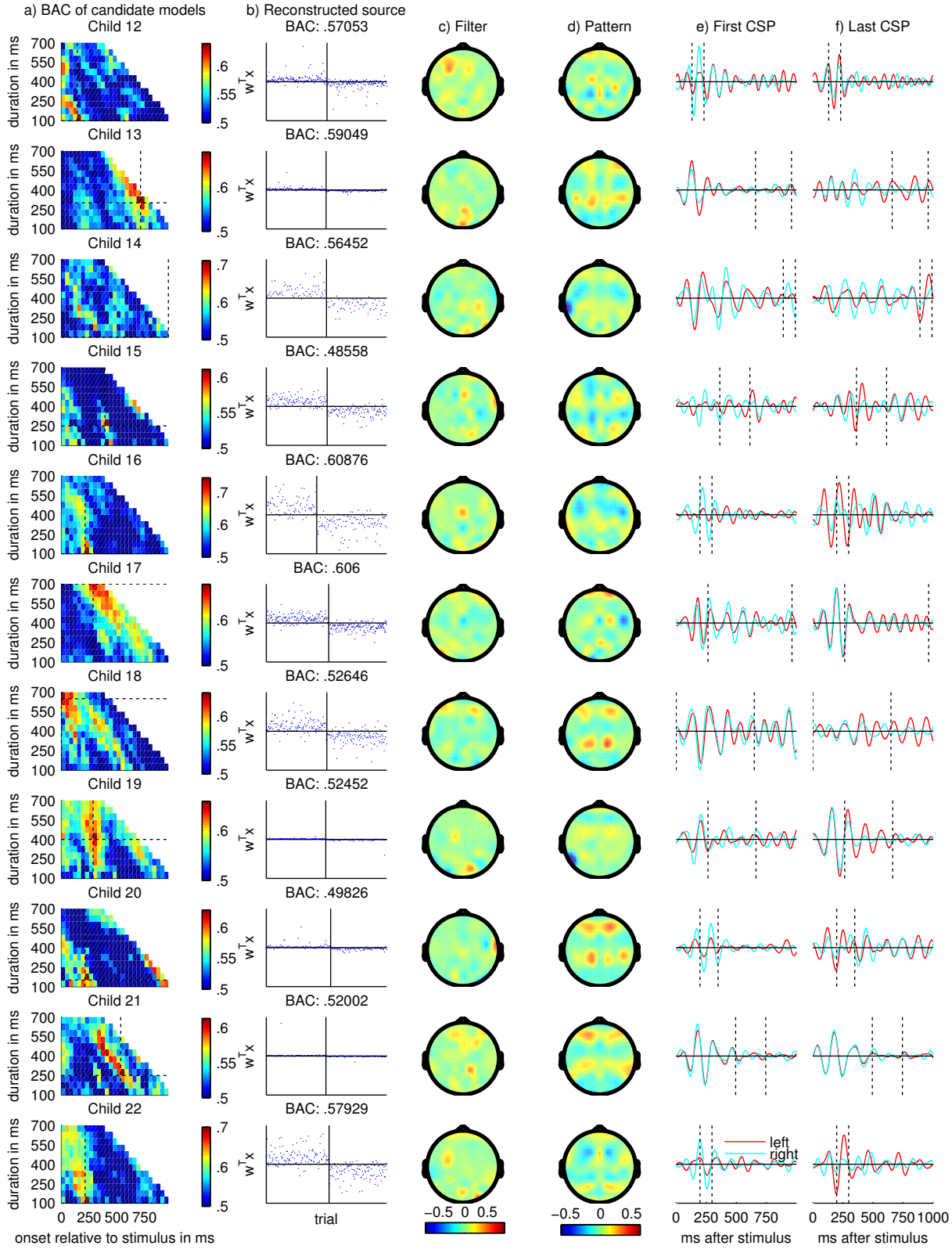
$$\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}, \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_n \end{bmatrix} \right).$$

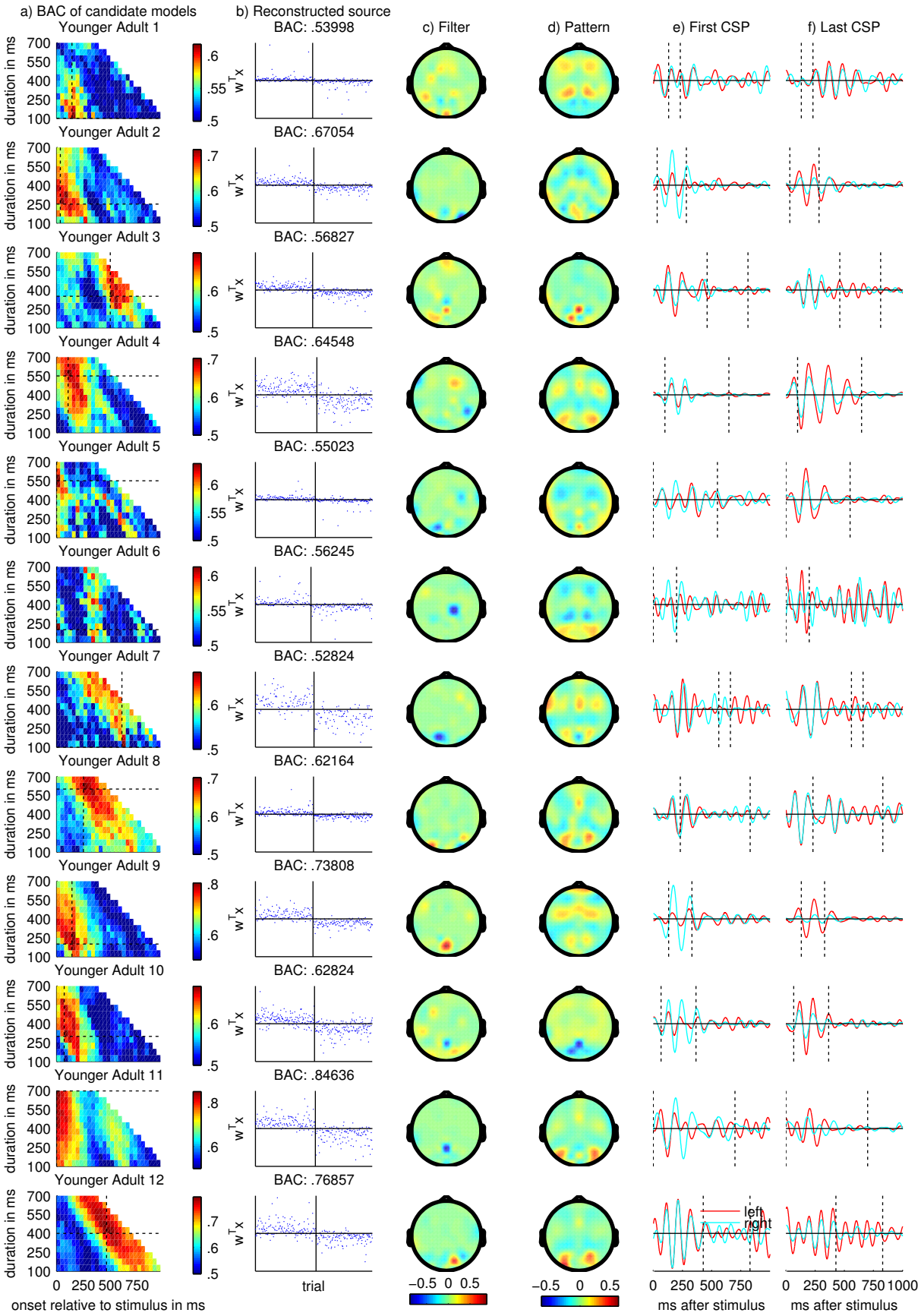
B. Person-Specific Results

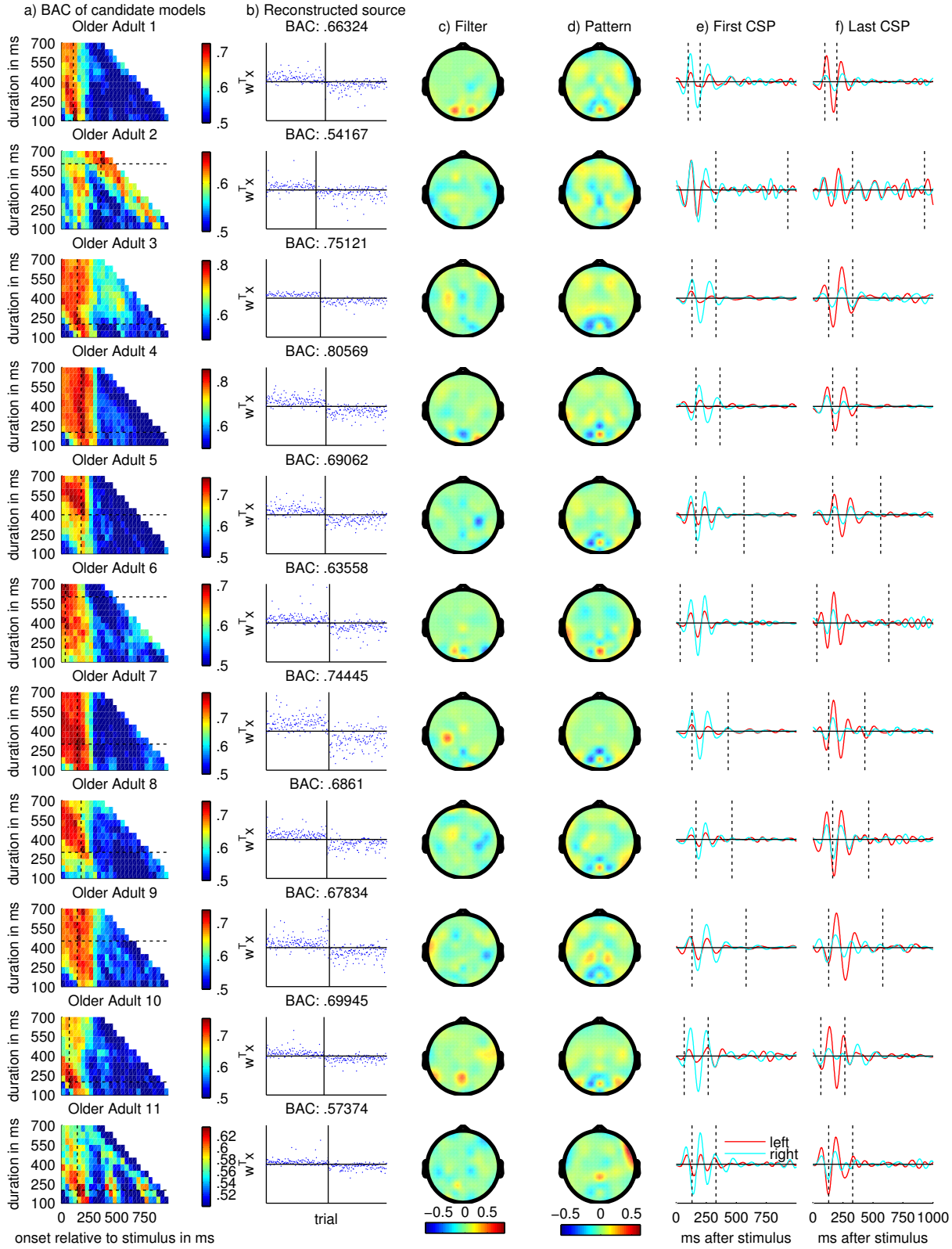
Here I show the person-specific results for each participant as obtained by the person-specific modeling approach introduced in Chapter 5. The following figures (B.1 for atttentional focus; B.2 for working memory (WM) load) are all in the same format: Column (a) shows the estimated balanced accuracy (BAC) for the different candidate models. The x-axis describes the onset and the y-axis the duration of the corresponding time window. Colors refer to the estimated BAC of the respective candidate model, with hot colors indicating higher BAC and cold colors lower BAC. The crosshair indicates the location of the selected model. Column (b) shows the estimated source component for each trial as reconstructed by the best estimated model, including the BAC of the best model. The trials are sorted by their true class. The vertical line separates the classes. The horizontal line marks 0. For a perfect classifier, the estimated source needs to be positive for all trials to the left and negative for all trials to the right of the vertical line. Column (c) shows the entries of the normalized filters, column (d) the normalized pattern, column (e) the mean time series for the first CSP filter for both classes, and column (f) the mean time series for the last CSP filter for both classes. The x-axis describes the time elapsed since the onset of the memory array. The vertical dotted lines indicate the selected time window. The horizontal line in columns (e) and (f) marks 0.

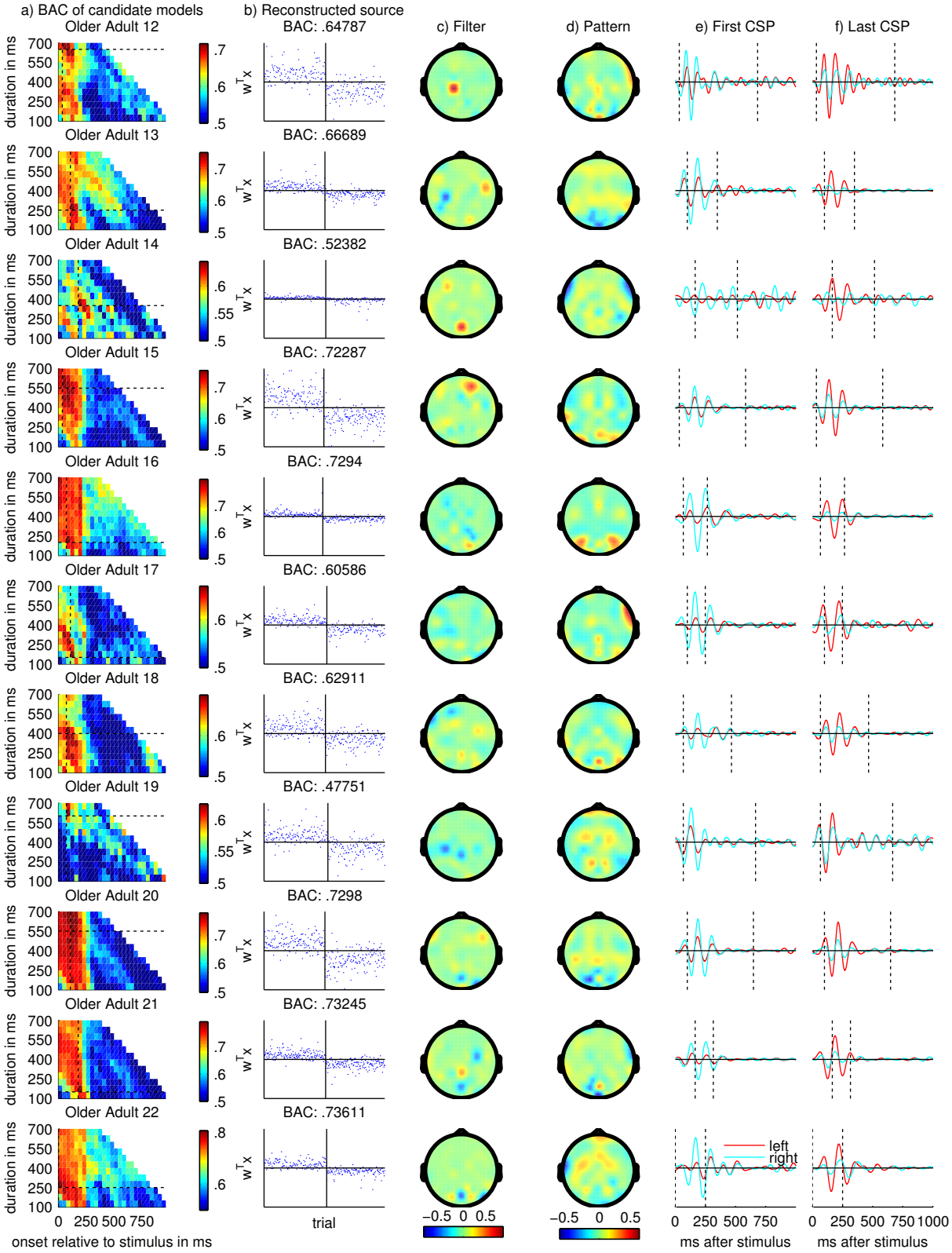
B.1. Attentional Focus



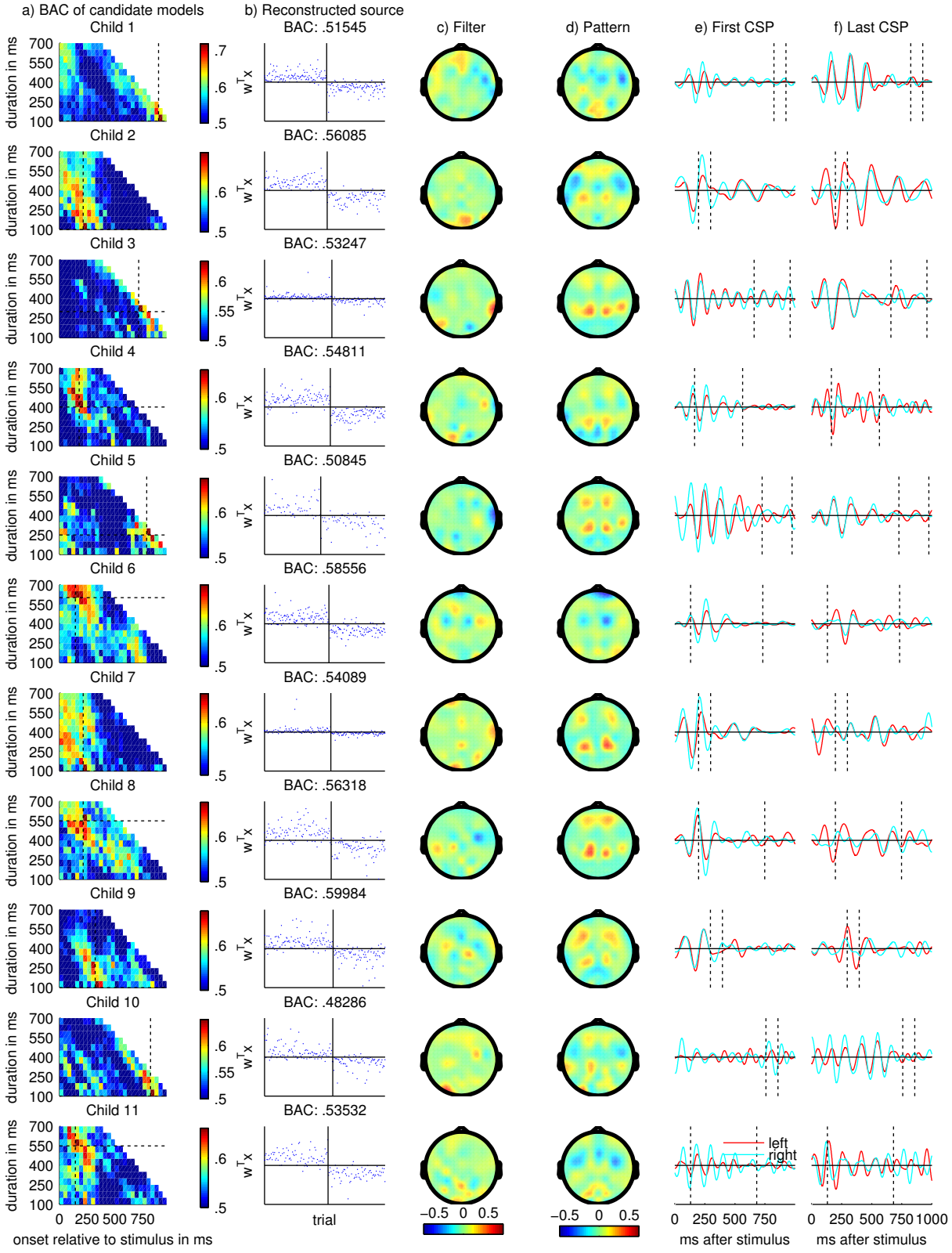

a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Child 12
BAC: .57053
Child 13
BAC: .59049
Child 14
BAC: .56452
Child 15
BAC: .48558
Child 16
BAC: .60876
Child 17
BAC: .606
Child 18
BAC: .52646
Child 19
BAC: .52452
Child 20
BAC: .49826
Child 21
BAC: .52002
Child 22
BAC: .57929
duration in ms
700
550
400
250
100
$w^T x$
.5
.55
.6
.7
0 250 500 750
onset relative to stimulus in ms
trial
−0.5 0 0.5
left
right
0 250 500 750
0 250 500 750 1000
ms after stimulus
ms after stimulus

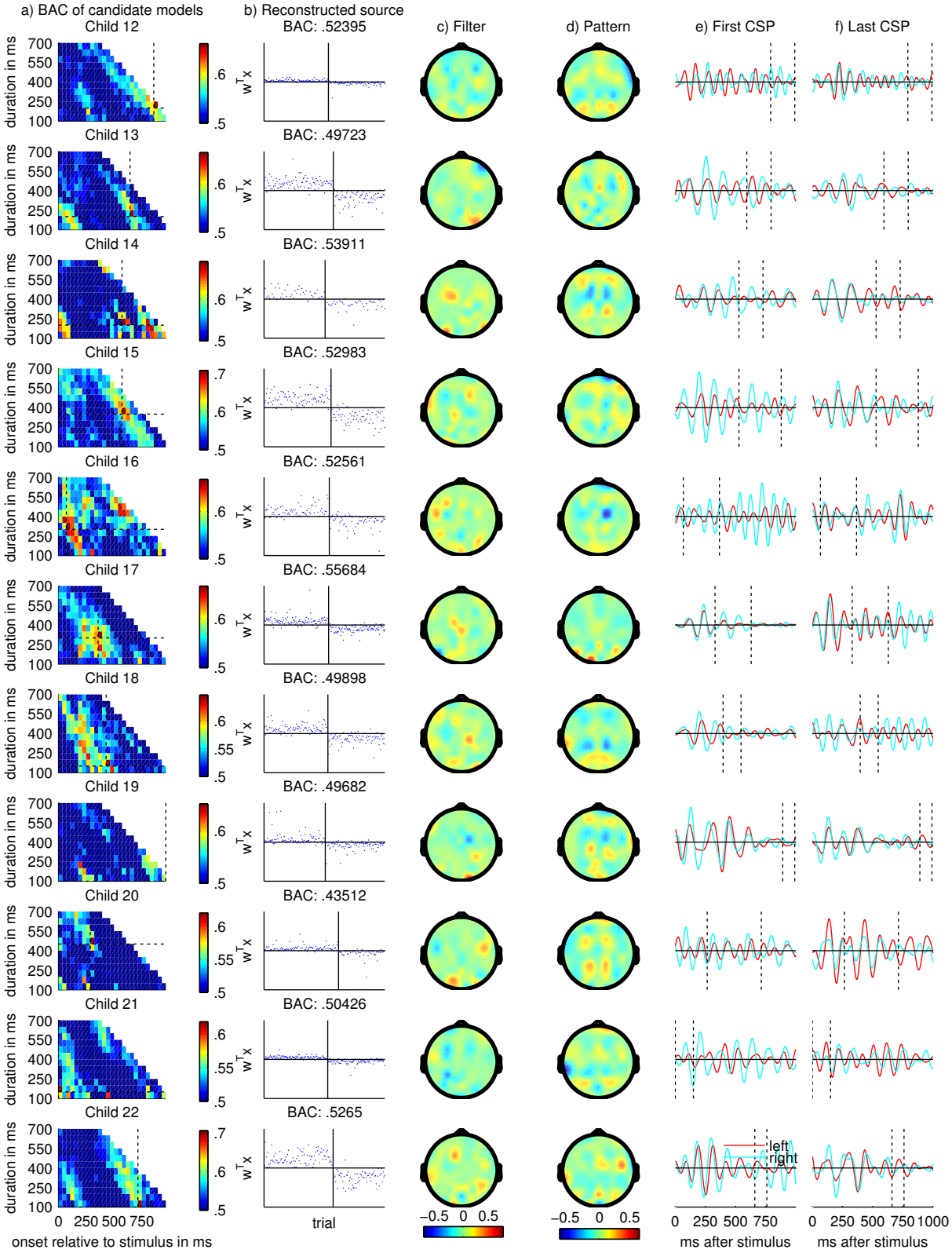

a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Younger Adult 1
BAC: .53998
Younger Adult 2
BAC: .67054
Younger Adult 3
BAC: .56827
Younger Adult 4
BAC: .64548
Younger Adult 5
BAC: .55023
Younger Adult 6
BAC: .56245
Younger Adult 7
BAC: .52824
Younger Adult 8
BAC: .62164
Younger Adult 9
BAC: .73808
Younger Adult 10
BAC: .62824
Younger Adult 11
BAC: .84636
Younger Adult 12
BAC: .76857
duration in ms
700
550
400
250
100
$w^T x$
trial
0 250 500 750
onset relative to stimulus in ms
−0.5 0 0.5
ms after stimulus
0 250 500 750 1000
left
right

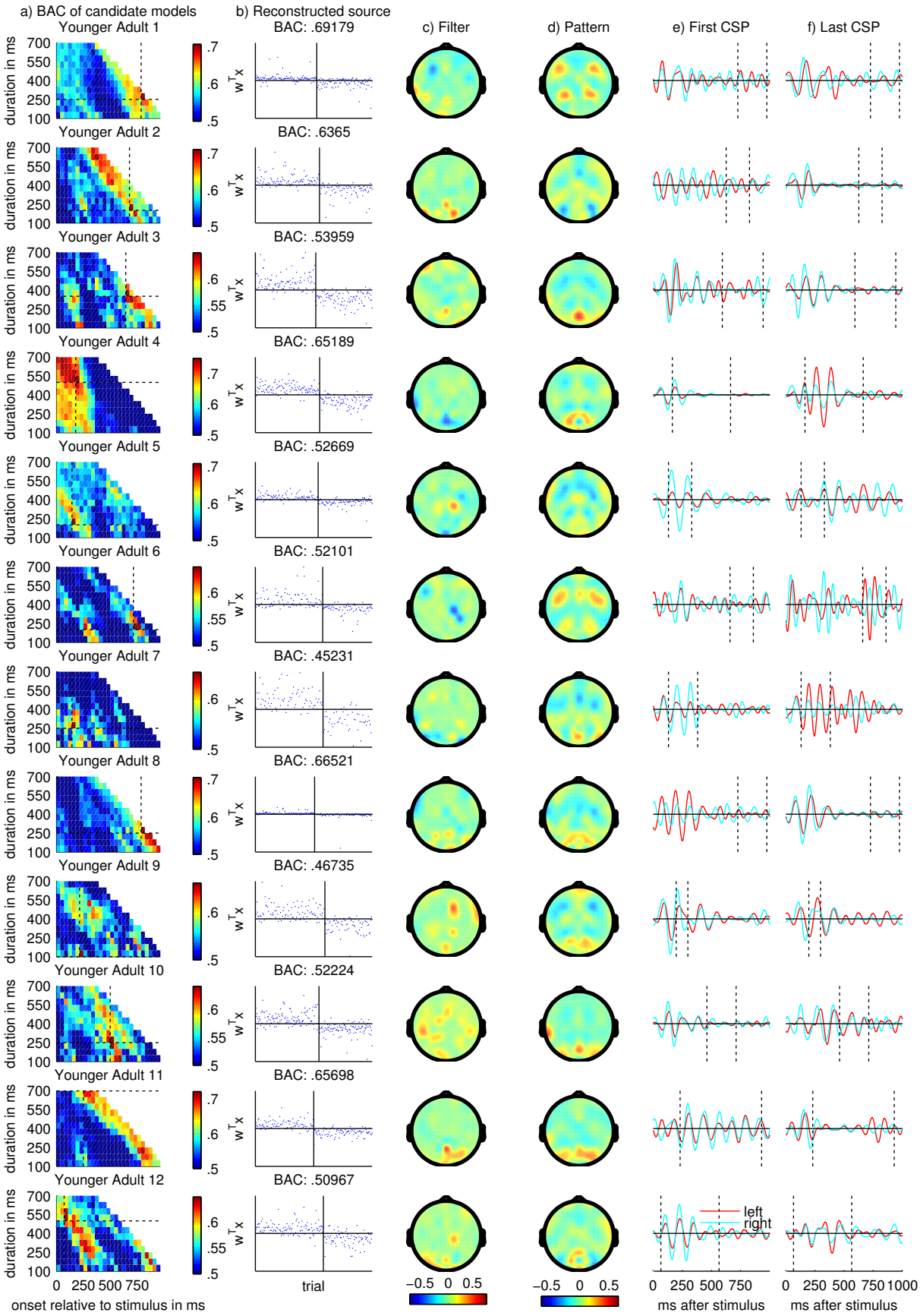

a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Older Adult 1
BAC: .66324
Older Adult 2
BAC: .54167
Older Adult 3
BAC: .75121
Older Adult 4
BAC: .80569
Older Adult 5
BAC: .69062
Older Adult 6
BAC: .63558
Older Adult 7
BAC: .74445
Older Adult 8
BAC: .6861
Older Adult 9
BAC: .67834
Older Adult 10
BAC: .69945
Older Adult 11
BAC: .57374
duration in ms
700
550
400
250
100
$w^T x$
trial
onset relative to stimulus in ms
0 250 500 750
−0.5 0 0.5
ms after stimulus
0 250 500 750 1000
left
right

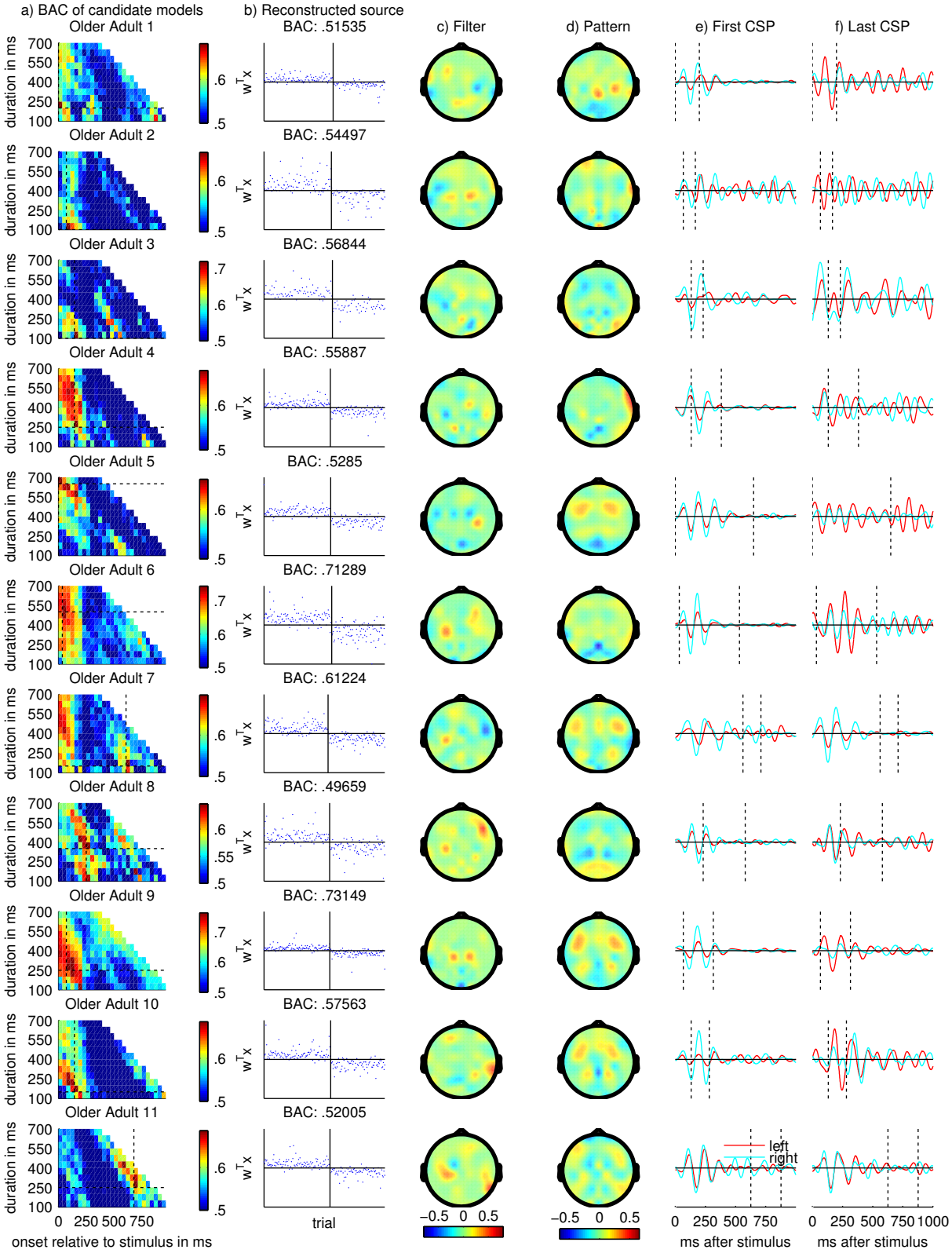

a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Older Adult 12
BAC: .64787
Older Adult 13
BAC: .66689
Older Adult 14
BAC: .52382
Older Adult 15
BAC: .72287
Older Adult 16
BAC: .7294
Older Adult 17
BAC: .60586
Older Adult 18
BAC: .62911
Older Adult 19
BAC: .47751
Older Adult 20
BAC: .7298
Older Adult 21
BAC: .73245
Older Adult 22
BAC: .73611
duration in ms
700
550
400
250
100
0 250 500 750
onset relative to stimulus in ms
trial
left
right
ms after stimulus
0 250 500 750 1000

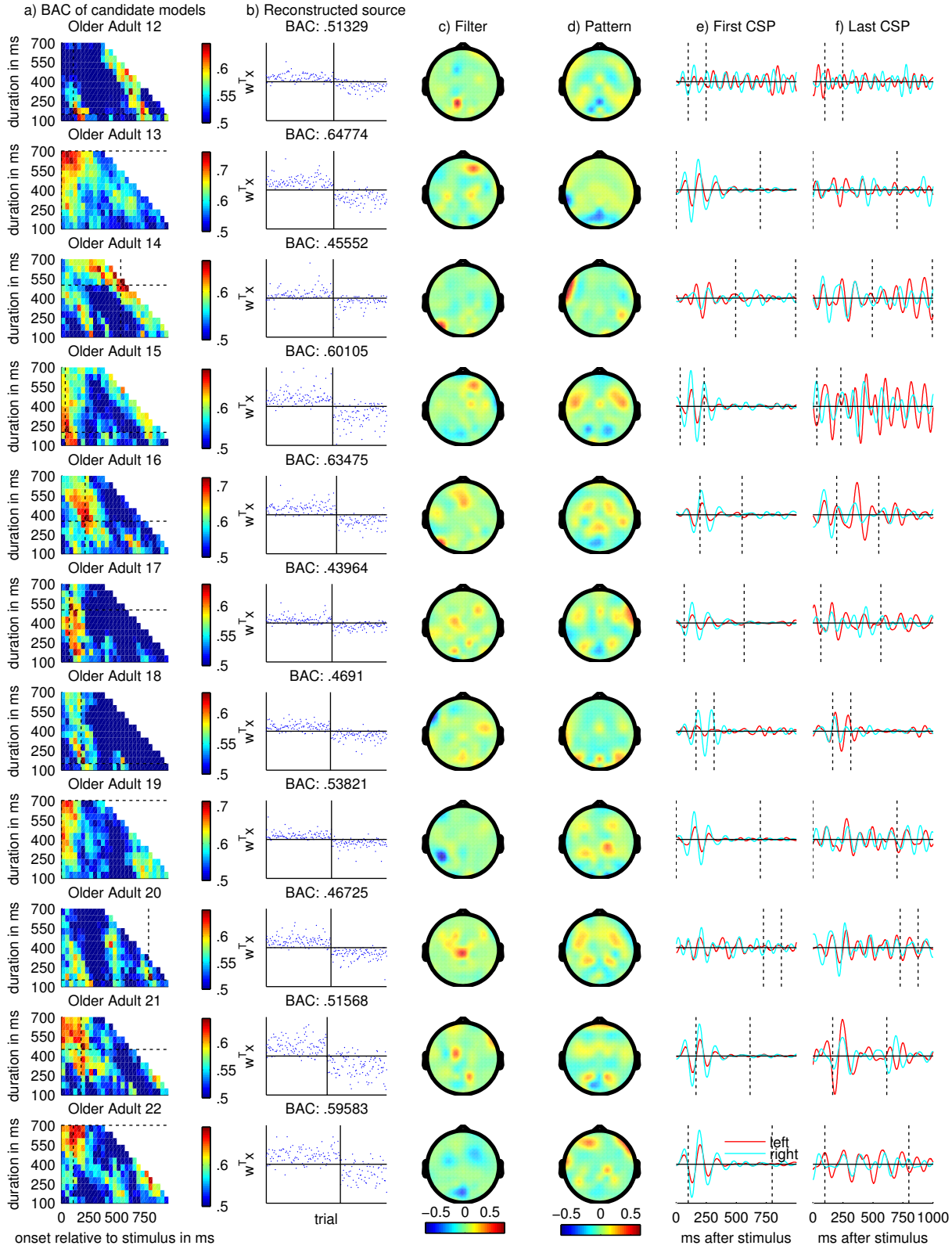
B.2. Working Memory Load




a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Child 12
BAC: .52395
Child 13
BAC: .49723
Child 14
BAC: .53911
Child 15
BAC: .52983
Child 16
BAC: .52561
Child 17
BAC: .55684
Child 18
BAC: .49898
Child 19
BAC: .49682
Child 20
BAC: .43512
Child 21
BAC: .50426
Child 22
BAC: .5265
duration in ms
700
550
400
250
100
$w^T x$
.7
.6
.55
.5
0 250 500 750
onset relative to stimulus in ms
trial
−0.5 0 0.5
left
right
0 250 500 750
ms after stimulus
0 250 500 750 1000
ms after stimulus


a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Younger Adult 1
BAC: .69179
Younger Adult 2
BAC: .6365
Younger Adult 3
BAC: .53959
Younger Adult 4
BAC: .65189
Younger Adult 5
BAC: .52669
Younger Adult 6
BAC: .52101
Younger Adult 7
BAC: .45231
Younger Adult 8
BAC: .66521
Younger Adult 9
BAC: .46735
Younger Adult 10
BAC: .52224
Younger Adult 11
BAC: .65698
Younger Adult 12
BAC: .50967
duration in ms
700
550
400
250
100
.7
.6
.55
.5
$w^T x$
0 250 500 750
onset relative to stimulus in ms
trial
−0.5 0 0.5
−0.5 0 0.5
left
right
0 250 500 750
ms after stimulus
0 250 500 750 1000
ms after stimulus


a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Older Adult 1
BAC: .51535
Older Adult 2
BAC: .54497
Older Adult 3
BAC: .56844
Older Adult 4
BAC: .55887
Older Adult 5
BAC: .5285
Older Adult 6
BAC: .71289
Older Adult 7
BAC: .61224
Older Adult 8
BAC: .49659
Older Adult 9
BAC: .73149
Older Adult 10
BAC: .57563
Older Adult 11
BAC: .52005
duration in ms
700
550
400
250
100
0 250 500 750
onset relative to stimulus in ms
trial
−0.5 0 0.5
ms after stimulus
0 250 500 750 1000
left
right


a) BAC of candidate models
b) Reconstructed source
c) Filter
d) Pattern
e) First CSP
f) Last CSP
Older Adult 12
BAC: .51329
Older Adult 13
BAC: .64774
Older Adult 14
BAC: .45552
Older Adult 15
BAC: .60105
Older Adult 16
BAC: .63475
Older Adult 17
BAC: .43964
Older Adult 18
BAC: .4691
Older Adult 19
BAC: .53821
Older Adult 20
BAC: .46725
Older Adult 21
BAC: .51568
Older Adult 22
BAC: .59583
duration in ms
700
550
400
250
100
0 250 500 750
onset relative to stimulus in ms
trial
−0.5 0 0.5
left
right
0 250 500 750
0 250 500 750 1000
ms after stimulus
ms after stimulus